

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

(повна назва інституту/факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«На правах рукопису»

УДК 004.021 _____

До захисту допущено:

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

«__» _____ 20__ р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютеризованих систем»**

зі спеціальності 121 «Інженерія програмного забезпечення»

**на тему: «Програмне та математичне забезпечення оптимізації
портфелю активів на ринку іноземних валют»**

Виконав (-ла):

студент (-ка) VI курсу, групи ПІ-92мп

Ясенова Анна Вадимівна _____

Науковий керівник:

ст.в. кафедри АСОІУ

Халус Олена Андріївна _____

Рецензент:

к.т.н., доц каф АУТС

Писаренко Андрій Володимирович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Автоматизованих систем обробки інформації і управління

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма - «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

«__» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Ясеновій Анні Вадимівній

1. Тема дисертації: «Математичне та програмне забезпечення оптимізації портфелю активів на ринку іноземних валют», науковий керівник дисертації Халус Олена Андріївна, старший викладач кафедри АСОІУ, затверджені наказом по університету від «26» жовтня 2020 р. №3132-с
2. Термін подання студентом дисертації: 11.12.2020 _____
3. Об'єкт дослідження: *процес формування оптимального портфелю на ринку іноземних валют.* _____
4. Вхідні дані: *постановка задачі* _____
5. Перелік завдань, які потрібно розробити: *налагодження ETL процесу, реалізація алгоритмів та дослідження їх ефективності, створення гнучкої архітектури застосунку, створення програмних інтерфейсів для передачі результатів роботи алгоритмів* _____
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: *Діаграма класів, діаграма варіантів використання* _____
7. Орієнтовний перелік публікацій: *Ясенова А.В., Халус О.А. Застосування алгоритмів кластеризації на ринку іноземних валют // V Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020» Матеріали конференції.* _____

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Графічний</i>	<i>доц. Ліщук К.І.</i>		

9. Дата видачі завдання 1.09.2020 _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	<i>Вивчення рекомендованої літератури</i>	<i>15.09.2020</i>	
2	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.09.2020</i>	
3	<i>Постановка та формалізація задачі</i>	<i>25.09.2020</i>	
4	<i>Моделювання ПЗ</i>	<i>1.10.2020</i>	
5	<i>Налагодження ETL процесу</i>	<i>6.10.2020</i>	
6	<i>Розробка алгоритмів та дослідження їх ефективності</i>	<i>15.10.2020</i>	
7	<i>Розробка архітектури ПЗ</i>	<i>1.11.2020</i>	
8	<i>Розробка і налагодження програмного забезпечення</i>	<i>20.11.2020</i>	
9	<i>Оформлення пояснювальної записки</i>	<i>25.11.2020</i>	
10	<i>Подання МД на попередній захист</i>	<i>27.11.2020</i>	
11	<i>Подання МД рецензенту</i>	<i>10.12.2020</i>	
12	<i>Подання МД на основний захист</i>	<i>16.12.2020</i>	

Студент

Анна ЯСЕНОВА

Науковий керівник

Олена ХАЛУС

РЕФЕРАТ

Актуальність теми: на даний момент не існує жодного сервісу, який дозволяє користувачу швидко дізнатися оптимальне співвідношення між активами в торговому портфелі, не дивлячись на те, що математично проблема давно вирішена.

Окрім цього, великою проблемою трейдерів-початківців є вибір активів, які входять до портфелю. На сьогоднішній день жоден з аналітичних сервісів ринку іноземних валют не надає користувачеві простого, і головне математично обгрунтованого способу для вибору активів до складу свого портфелю.

Мета дослідження: Основною метою є дослідження та розробка архітектури програмного забезпечення для зменшення витрат часу користувача на створення портфелю активів шляхом поєднання в одному застосунку алгоритмів кластеризації та оптимізації.

Для реалізації поставленої мети сформульовані **наступні завдання:**

- налагодження ETL процесу в системі;
- програмна реалізація алгоритмів;
- дослідження і порівняння швидкодії реалізованих алгоритмів;
- побудова гнучкої архітектури десктопного застосунку;
- створення програмних інтерфейсів для передачі результатів роботи

алгоритму стороннім джерелам.

Об'єкт дослідження: процес розробки програмного забезпечення формування оптимального портфелю на ринку іноземних валют.

Предмет дослідження: алгоритми кластеризації та методи оптимізації, програмні бібліотеки алгоритмів оптимізації та кластеризації, шляхи поєднання кластеризації та математичної оптимізації в межах одного програмного застосунку.

Наукова новизна результатів магістерської дисертації полягає в тому, що запропоновано архітектурне рішення для побудови програмного забезпечення для створення торгового портфелю, яке на відміну від інших

надає користувачеві очікуваний результат при мінімальних затратах часу та кількості необхідних дій для початку роботи. Результат досягнутий шляхом розробки модернізованого алгоритму оптимізації.

Практичне значення отриманих результатів полягає в тому, що реалізовані методи поєднані в межах одного застосунку і максимально прості у використанні для користувача. Також реалізовано API-інтерфейс, за допомогою якого результати роботи алгоритмів можуть з легкістю отримувати і застосовувати сторонні сервіси.

Зв'язок з науковими програмами, планами, темами: робота виконувалась на кафедрі автоматизованих систем обробки інформації і управління Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Публікації: Наукові положення дисертації опубліковані в Ясенова А.В. Застосування алгоритмів кластеризації на ринку іноземних валют/ А.В. Ясенова, О.А. Халус // Матеріали V всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р.

КЛЮЧОВІ СЛОВА: КЛАСТЕРИЗАЦІЯ, ОПТИМАЛЬНИЙ ТОРГОВИЙ ПОРТФЕЛЬ, ФОРЕКС, ОПТИМІЗАЦІЯ

ABSTRACT

Topicality: today there is no service which allows quickly find out optimal weights for trading portfolio components, despite the fact that mathematically the problem has long been solved.

It is also very difficult for novice traders to choose the assets that are part of the portfolio. Today, none of the analytical services of the foreign exchange market provides the user with a simple and, most importantly, mathematically reliable way to compose trading portfolio.

The aim of the study: the main target is to research and develop software architecture for decreasing the time spent on the portfolio creation by combining in one application clustering and optimization algorithms.

To achieve this goal, the **following tasks** were formulated:

- debug the ETL process;
- implementation of algorithms;
- compare efficiency of implemented algorithms;
- build flexible infrastructure;
- create API interfaces to transfer results of work to internal sources.
- create interfaces for receiving results of work of algorithms.

Object of research: the process of developing software for composing the optimal portfolio in the foreign exchange market.

Subject of research: clustering algorithms and optimization methods, software libraries of optimization and clustering algorithms, ways to combine clustering and mathematical optimization within one software application.

The scientific novelty of the results of the master's dissertation is that for the first time proposed architecture decision for building software for composing a trading portfolio, which, unlike others, provides the user with the expected result with minimal time and the number of necessary actions to get started. The result was achieved by developing an upgraded optimization algorithm.

The practical value of the obtained results is that the implemented methods are combined within one application and are as easy to use for the user. Also implemented API-interface, through which the results of the algorithms can easily receive and use third-party services.

Relationship with working with scientific programs, plans, topics: work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky».

Publications: Scientific provisions of the dissertation published in Yasenova A.V. The application of clustering methods on the foreign exchange market / A.V.Yasenova, O.A. Khalus // Proceedings of the Fifth All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" (ISTU- 2020) - Kyiv: NTUU "KPI them. Igor Sikorsky", November 26-27, 2020.

KEYWORDS: CLUSTERING, OPTIMAL TRADING PORTFOLIO, FOREX, OPTIMIZATION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП.....	10
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	14
1.3.1 <i>Аналіз відомих технічних рішень</i>	<i>14</i>
1.3.2 <i>Аналіз відомих програмних продуктів.....</i>	<i>15</i>
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	17
1.4.1 <i>Розроблення функціональних вимог.....</i>	<i>18</i>
1.4.2 <i>Розроблення нефункціональних вимог</i>	<i>26</i>
1.5 ПОСТАНОВКА ЦІЛЕЙ ТА ЗАДАЧ ДОСЛІДЖЕННЯ	27
1.6 ВИСНОВКИ ДО РОЗДІЛУ	27
2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	28
2.1 АЛГОРИТМИ КЛАСТЕРИЗАЦІЇ	28
2.1.1 <i>KMeans.....</i>	<i>28</i>
2.1.2 <i>DBSCAN.....</i>	<i>31</i>
2.2 АЛГОРИТМИ ОПТИМІЗАЦІЇ	34
2.3 ВИСНОВКИ ДО РОЗДІЛУ	37
3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	38
3.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	38
3.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	39
3.3 АНАЛІЗ БЕЗПЕКИ ДАНИХ	45
3.4 КЕРІВНИЦТВО КОРИСТУВАЧА	46
3.5 ВИСНОВКИ ДО РОЗДІЛУ	48
4 РЕЗУЛЬТАТИ ЕКСПЕРЕМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ	49
4.1 МЕТА ТА ПОРЯДОК ДОСЛІДЖЕНЬ.....	49
4.2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ОПТИМІЗАЦІЇ.....	49
4.3 ДОСЛІДЖЕННЯ ШВИДКОДІЇ АЛГОРИТМІВ ОПТИМІЗАЦІЇ.....	50

4.4	Висновки до розділу	52
5	РОЗРОБКА СТАРТАП ПРОЕКТУ.....	53
5.1	ОПИС ОСНОВНОЇ ІДЕЇ ПРОЕКТУ	53
5.2	ТЕХНОЛОГІЧНИЙ АУДИТ ІДЕЇ ПРОЕКТУ	55
5.3	АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ ЗАПУСКУ СТАРТАП-ПРОЕКТУ	56
5.4	РОЗРОБЛЕННЯ РИНКОВОЇ СТРАТЕГІЇ ПРОЕКТУ	63
5.5	РОЗРОБЛЕННЯ МАРКЕТИНГОВОЇ ПРОГРАМИ ПРОЕКТУ	66
5.6	Висновки до розділу	68
	ВИСНОВКИ	70
	ПЕРЕЛІК ПОСИЛАНЬ.....	71
	ДОДАТОК А ТЕКСТ ПРОГРАМНОГО КОДУ	72
	ДОДАТОК Б СХЕМА СТРУКТУРНА ВАРІАНТІВ ВИКОРИСТАННЯ.....	92
	ДОДАТОК В ДІАГРАМА КЛАСІВ.....	94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Форекс – це ринок, на якому відбуваються різноманітні операції з обміну валют.

Треjder – людина, що здійснює операції на форексі.

Активи на ринку форекс – це валютні пари, індекси, акції, цінні папери.

Ануалізована величина – величина, приведена до періоду 1 рік, середньорічне значення величини, або середнє геометричне значення цієї міри.

Перфоманс – величина, що характеризує дохідність активу/трейдера.

Коефіцієнта шарпа – показник ефективності активу/трейдера, що враховує очікуваний дохід і очікувані ризики.

Кластеринг – це процес розділення множини об'єктів на підмножини за логікою, що задається алгоритмом.

ВСТУП

Ринок іноземних валют розвивається доволі довгий час і зараз, за допомогою сучасних технологій, будь-хто може торгувати не виходячи з дому, необхідний лише комп'ютер чи телефон і вихід в інтернет.

Так як і для інших сфер людської діяльності, для ринку іноземних валют розробляється велика кількість різноманітних програмних продуктів: аналітичні сервіси, торгові платформи, боти, розповсюджувачі даних та інші. Разом з цим, форекс не може існувати без даних. Основні дані, якими оперує біржа – це ціни на активи, точкові конверсії між валютами, інформація про торгову історію клієнта. Також є неосновні типи наборів даних, які виникли в процесі розвитку інших сфер, наприклад текстові набори даних про події в світі, або сентимент аналіз постів і коментарів у соціальних мережах.

Зазвичай програмне забезпечення для ринку іноземних валют призначене для надання доступу до певної інформації або створення умов для торгівлі. Доволі великою проблемою є те, що для початку користування таким сервісом користувач має розуміти як працює ця сфера та бути компетентним в питаннях торгівлі хоча б на базовому рівні. І лише протягом останніх кількох років почали з'являтися сервіси, які мають значно нижчий поріг входу, або надають можливість для швидкого вивчення базових понять.

Проблема оптимізації портфелю активів на ринку іноземних валют математично вирішена багато років тому, і має нескладне формулювання: мінімізація цільової функції з обмеженнями. Проте, на даний момент не реалізовано жодного застосунку, де користувач може з легкістю отримати результати роботи такого оптимізатора без докладання зайвих зусиль. Варто зазначити, що для вирішення задачі необхідно правильно обрати цільову функцію, обмеження для неї, дослідити поведінку в крайніх ситуаціях, обрати набір даних – все це є складною послідовністю дій для людини, яка не має досвіду в програмуванні, компетенції в області трейдингу та математичній оптимізації.

Тому **метою дослідження** є дослідження та розробка архітектури програмного забезпечення для зменшення витрат часу користувача на створення портфелю активів шляхом поєднання в одному застосунку алгоритмів кластеризації та оптимізації.

Для реалізації поставленої мети сформульовані **наступні завдання**:

- налагодження ETL процесу в системі;
- програмна реалізація алгоритмів;
- дослідження і порівняння швидкодії реалізованих алгоритмів;
- побудова гнучкої архітектури десктопного застосунку;
- створення програмних інтерфейсів для передачі результатів роботи алгоритму стороннім джерелам.

Об'єктом дослідження є процес розробки програмного забезпечення формування оптимального портфелю на ринку іноземних валют, а **предметом дослідження** є алгоритми кластеризації та методи оптимізації, програмні бібліотеки алгоритмів оптимізації та кластеризації, шляхи поєднання кластеризації та математичної оптимізації в межах одного програмного застосунку.

Наукова новизна результатів магістерської дисертації полягає в тому, що запропоновано архітектурне рішення для побудови програмного забезпечення для створення торгового портфелю, яке на відміну від інших надає користувачеві очікуваний результат при мінімальних затратах часу та кількості необхідних дій для початку роботи. Результат досягнутий шляхом розробки модернізованого алгоритму оптимізації.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Ринок іноземних валют – це складна система в якій співіснують трейдери, брокери, біржі, посередники і т.д. Світ технологій невпинно розвивається, відповідно й для форексу з часом з'явилася дуже велика кількість програмного забезпечення: від рекламних сайтів компаній, основною метою яких є привабити потенційних клієнтів, до різноманітних порадників та повноцінних торгових систем, які мінімізують необхідність контролювати ситуацію на ринку і торгують замість людей.

Технології, які використовує трейдер на ринку іноземних валют значною мірою впливають на результат його торгівлі: важливим фактором є призначення програмного забезпечення, якість реалізації та інформація, яку використовує програма для своєї роботи.

Варто також згадати той факт, що в нашій країні та деяких країнах-сусідах ставлення до форексу скоріше негативне, ніж навпаки, а все тому, що в час кризи з'явилася велика кількість шахраїв, які заохочували людей вкладати свої гроші в свій «бізнес» і програвали ці гроші. Хотілося б наголосити, що наше дослідження несе суто науковий характер, дана сфера є хорошим підґрунтям для дослідження, так як в інтернет ресурсах у вільному доступі є багато різноманітної інформації для досліджень, що часто є великою проблемою.

1.2 Змістовний опис і аналіз предметної області

Програмне забезпечення для ринку іноземних валют можна поділити на наступні типи:

- прості лендингові інтернет-сторінки – це одно або багатосторінкові сайти, які зазвичай використовуються компаніями для того, щоб познайомити клієнта з компанією чи продуктом;

– торгові платформи – це таке програмне забезпечення, яке надає повний функціонал, що необхідний для процесу трейдингу, а саме можливість здійснювати торгові операції (покупка і продаж активів, заведення і виведення коштів), доступ до актуальних курсів валют та їх зміни за різні періоди часу, динамічна візуалізація курсу валюти. Це може бути будь-який тип застосунку: веб-сервіс, мобільний додаток чи комп'ютерна програма. Розширені можливості такого забезпечення включають в себе можливість використовувати торгових порадників, ботів і сигнали. При реалізації таких застосунків важливо розуміти, що найбільш важливою вимогою до них є насамперед якість даних і відсутність затримок при виконанні операцій[1];

– торгові сигнали – це такі собі програми-інформатори, основним призначенням яких є повідомляти трейдеру про певні події на ф'ючерс. Зазвичай вони працюють всередині торгової платформи або веб-сервісу, на основі заданого алгоритму визначають, чи настала необхідна подія (наприклад, курс валюти досягнув найбільшого значення за останній тиждень) та інформують про це трейдера. Також існують вбудовані сигнали – по факту це інструментарій для брокера, за допомогою якого він може доносити інформацію до користувача. Важливою вимогою до такого типу програмного забезпечення є актуальність сигналів, так як вони втрачають цінність майже миттєво. Звідси наступна вимога – швидкість передачі даних має бути на високому рівні, тобто проміжок часу між генерацією сигналу і отриманням нотифікації користувачем має бути мінімальним;

– порадики та індивідуальні ф'ючерс-роботи – це такі програми, що схожі на торгові сигнали, тобто вони також аналізують інформацію, що приходить з ринку, проте на відміну від сигналів, де людина сама приймає рішення чи виконувати якісь дії на ринку, роботи повністю виконують весь процес торгівлі самостійно. Для того, щоб така програма запрацювала необхідно лише створити торговий рахунок, переконатися в наявності коштів і запустити алгоритм. Основна відмінність між порадиками та ф'ючерс-роботами

в тому, що перші зазвичай є безкоштовними і загальнодоступними, а другі виконуються на замовлення або для продажу. Тобто по факту ці програми мають одне і те ж саме призначення і принцип роботи. Такі програми надають можливість трейдерам реалізувати власні алгоритми, випробувати власні припущення і автоматизувати процес торгівлі. Проте необхідно враховувати ризики, які виникають під час використання такого способу торгівлі: робот не зможе відреагувати на критичну ситуацію, якщо це не закладено його алгоритмом, відповідно якщо торгівля зовсім не контролюється людиною, ризики втратити всі гроші зростають. Основною вимогою до таких програмних продуктів є здатність заробляти гроші при мінімальних ризиках і просадках.

Зокрема недоброчесні компанії використовують наступну схему: через лендинг-сторінку вони приваблюють клієнта на свій сайт, потім, з використанням телефонії вони стимулюють клієнта зробити депозит на свій рахунок і вже після цього програють його гроші на торговій платформі. Іноді для торгівлі використовуються порадики або боти.

1.3 Аналіз успішних IT-проектів

1.3.1 Аналіз відомих технічних рішень

Чи не найважливішою умовою досягнення успіху для будь-якого продукту ринку іноземних валют є якість даних, які він використовує. Адже яким би хорошим не був графічний інтерфейс чи реклама, якщо сервіс надає неточні дані, рано чи пізно клієнт збанкрутує або перейде до кращого конкурента. Зараз будь-хто з доступом до мережі Інтернет може отримати ціни різних активів (валютні пари, індекси, акції, дорогоцінні метали і т.д.). Багато сервісів, такі як Oanda, ExchangeRatesAPI та Fixer.io надають можливість отримати безкоштовну підписку, що надає право завантажувати дані з сервісу за допомогою REST API.

Варто відзначити, що така безкоштовна підписка зазвичай є дуже обмеженою, наприклад для даного дослідження використовується програмний

інтерфейс сервісу Fixer.io, він надає можливість завантажувати лише денні ціни активів і накладає обмеження на 1000 запитів до API на місяць. Для багатьох досліджень, в яких необхідно дослідити прості залежності, цього достатньо. Проте для комерційного використання така підписка не підійде. Разом з цим, якщо дослідження чи продукт передбачає знаходження складніших залежностей або прогнозування на коротших проміжках часу, то потрібна більша частота даних (наприклад погодинні, а в ідеалі, дані з хвилинною частотою). Варто відмітити, що сервіси надають такі програмні інтерфейси, для яких зручно писати власні обгортки на будь-якій мові програмування, що значно спрощує впровадження таких джерел даних в продукт.

Для того, щоб створити аналітичну частину продукту необхідно використовувати призначені для цього мови програмування. Стандартними інструментами для статистичних досліджень є Python, MATLAB, R. З їх допомогою можна дослідити базові та розширені статистики різних наборів даних, візуалізувати дані, натренувати, відкалібрувати та задеплоїти модель. В даному дослідженні використано мову програмування Python, так як з її допомогою виконана не лише аналітична частина продукту, а й технічна, а саме обгортка до API-інтерфейсу, візуальна частина додатку.

1.3.2 Аналіз відомих програмних продуктів

Важливою частиною ринку іноземних валют є, власне, процес торгівлі. Можливість торгівлі надається брокером, а вихід на біржу, тобто можливість здійснювати транзакції, є в торгових платформ.

Брокер – це компанія-посередник між продавцями і покупцями валют. Зараз брокери будують процес таким чином, щоб для клієнта функція «посередництва» була непомітною і не завдавала зайвих незручностей. Існує велика кількість таких компаній, лідери в галузі постійно змінюються, так як в залежності від зміни регуляцій компанії можуть припиняти свою діяльність.

Представниками брокерських компанії на ринку іноземних валют є Альпарі, АхіTrader, Investing та інші.

Торгова платформа – це комп’ютерна система, яка використовується для розміщення транзакцій на купівлю/продаж будь-яких активів, доступних на ринку іноземних валют. Такі програмні засоби використовуються різними брокерами і дилерами для реалізації процесу електронної торгівлі. Найбільш відома торгова платформа – це MetaTrader, менш відомими є OnlineBroker, QUIK. MetaTrader надає можливість підключення новин з різних джерел, дає доступ для маржинальної торгівлі з кредитним плечем і дозволяє встановлювати автоматичний контроль лімітів. Також надається можливість створення власних додатків як для візуального відображення, так і для автоматичної торгівлі за допомогою С-подібної мови MQL. Також можна використовувати власні джерела даних для аналізу і бектестингу стратегій (Бектест – перевірка роботи торгової системи на історичних даних, по факту – це спроба відтворити роботу радника чи робота і зрозуміти, чи був би він ефективний в минулому). Великий недолік даного програмного продукту в тому, що десктопний додаток розроблений лише для Windows OS, тому всі користувачі операційних систем Unix сімейства мають використовувати веб-версію платформи, що має обмежений функціонал.

Аналіз даних продовжує набувати популярності, тому природньою була поява різних аналітичних систем і для ринку форекс. Однією з перших таких платформ myfxbook.com. Вона надає можливість проаналізувати ефективність торгівлі рахунку, порівняти свої результати з іншими трейдерами, візуалізувати дані. Також даний сайт призначений для формування спільноти трейдерів, надає можливість поширювати свої результати, розміщувати оголошення з необхідністю виконання радників чи робіт. Даний сайт має застарілий та інтуїтивно незрозумілий дизайн, функціонал несистематизований, тому необхідні розділи сервісу важко знайти.

ForexAnalytix – це більш нова платформа для надання аналітичних послуг для ринку форекс, що дає доступ до різних візуалізацій і показників для найбільш популярних активів на ринку, різноманітних відеоресурсів та матеріалів, джерел новин та даних. Є можливість використання веб-застосунку, комп'ютерного додатку та мобільної версії. Проте сервіс не надає безкоштовної підписки.

Якщо ж є необхідність використання такого сервісу на короткий період часу, можна спробувати Autochartist – це менш популярна платформа, проте вона надає можливість безкоштовного 14-денного періоду користування, активно розвивається і покриває увесь необхідний спектр функціональних потреб.

PsyQuation - це австралійська платформа для ринку іноземних валют. Вона розробляється як інструмент для кількісного та поведінкового аналізу трейдерів та активів на форексі. Власне, після реєстрації на сайті і підключення власного рахунку є доволі велика кількість функціоналу, що доступна безкоштовно: аналіз ефективності торгівлі, різноманітні статистики, інтерактивні засоби для торгівлі.

1.4 Аналіз вимог до програмного забезпечення

Математичне та програмне забезпечення повинне містити такі типи користувачів як адміністратор і користувач.

Адміністратор – це користувач з особливими правами, має доступ до сервера на якому налагоджений процес завантаження та збереження даних, а також має можливість змінювати гіперпараметри алгоритмів системи.

Користувач має доступ до функціоналу сервісу: запуск кластерингу для виявлення груп активів зі схожою поведінкою і запуск оптимізації для визначення оптимального співвідношення активів в портфелі.

В застосунку повинні бути реалізовані наступні функції:

- кластеринг валют;

- оптимізація портфелю активів.

1.4.1 Розроблення функціональних вимог

Схема структурна варіантів використання наведена в додатках до магістерської дисертації.

В системі передбачено наступні варіанти використання, описані у таблицях 1.1 – 1.11:

Таблиця 1.1 - Варіант використання UC001

Назва	Зміна гіперпараметрів алгоритмів
Опис	Адміністратор змінює гіперпараметри алгоритмів регулярно і за запитом
Учасники	Адміністратор
Передумови	Алгоритми потребують рекалькуляції параметрів щомісяця, в залежності від умов ринку перерахунок іноді потрібен частіше
Постумови	Гіперпараметри алгоритмів змінені
Основний сценарій	Адміністратор вручну змінює гіперпараметри алгоритмів, що доступні в застосунку

Таблиця 1.2 - Варіант використання UC002

Назва	Додавання методу оптимізації (алгоритму кластеризації) до списку доступних в застосунку
Опис	Адміністратор може додавати методи до списку доступних для використання алгоритмів в застосунку
Учасники	Адміністратор
Передумови	Налагоджений і протестований новий алгоритм, наявні всі необхідні дослідження, готові скрипти для формування датасету

Продовження таблиці 1.2

Постумови	Алгоритм доступний для використання в застосунку
Основний сценарій	Адміністратор вручну додає новий метод до списку доступних алгоритмів в застосунку

Таблиця 1.3 - Варіант використання UC003

Назва	Видалення методу оптимізації (алгоритму кластеризації) зі списку доступних в застосунку
Опис	Адміністратор може видаляти методи зі списку доступних для використання алгоритмів в застосунку
Учасники	Адміністратор
Передумови	Виникли проблеми в роботі алгоритму, необхідні додаткові дослідження, неточності не можуть бути виправлені одразу і потребують часу
Постумови	Алгоритм не доступний для використання в застосунку
Основний сценарій	Адміністратор вручну видаляє алгоритм зі списку доступних в додатку

Таблиця 1.4 - Варіант використання UC004

Назва	Вибір довжини історичного періоду для аналізу
Опис	Користувач може обирати довжину часового періоду, який буде використаний для обчислення вхідних параметрів алгоритмів
Учасники	Користувач

Продовження таблиці 1.4

Передумови	Користувач перейшов на сторінку кластеризації активів або на сторінку оптимізації інвестиційного портфелю
Постумови	Довжина часового періоду (в днях) обрана
Основний сценарій	Користувач переходить на сторінку кластеризації або оптимізації і обирає кількість днів якій рівна довжина часового проміжку для аналізу

Таблиця 1.5 - Варіант використання UC005

Назва	Вибір алгоритму кластеризації
Опис	Користувач може вводити необхідну для конвертації кількість валюти
Учасники	Користувач
Передумови	Користувач перейшов на сторінку розрахунку вартості конвертації
Постумови	Кількість валюти введена
Основний сценарій	Користувач переходить на сторінку розрахунку вартості конвертації і вводить кількість валюти для розрахунку вартості конвертації

Таблиця 1.6 - Варіант використання UC006

Назва	Перегляд результатів кластеризації в текстовому вікні
Опис	Результати роботи алгоритму кластеризації активів відображаються в окремому текстовому вікні
Учасники	Користувач

Продовження таблиці 1.6

Передумови	Користувач перейшов на сторінку кластеризації активів портфелю активів і обрав період часу для аналізу
Постумови	Результат роботи алгоритму кластеризації відображений в текстовому вікні
Основний сценарій	Користувач обрав всі вхідні параметри і натиснув кнопку “Run Clustering”

Таблиця 1.7 - Варіант використання UC007

Назва	Вибір методу оптимізації
Опис	Користувач може обирати метод для оптимізацію портфелю активів
Учасники	Користувач
Передумови	Користувач перейшов на сторінку оптимізації портфелю активів
Постумови	Метод оптимізації портфелю активів обраний
Основний сценарій	Користувач переходить на сторінку оптимізації портфелю активів і обирає метод оптимізації з представленого списку

Таблиця 1.8 - Варіант використання UC008

Назва	Вибір активів, що входять до портфелю
Опис	Користувач може обирати активи, які будуть входити в портфель, який необхідно оптимізувати
Учасники	Користувач
Передумови	Учасник знаходиться на сторінці оптимізації портфелю активів
Постумови	В необхідному полі введена кількість валюти

Продовження таблиці 1.8

Основний сценарій	Користувач переходить на сторінку оптимізації портфелю активів і обирає зі списку валюти, які входять до портфелю
-------------------	---

Таблиця 1.9 - Варіант використання UC009

Назва	Введення кількості грошей, які будуть інвестовані в портфель
Опис	Користувач може вводити кількість грошей, які будуть інвестовані в портфель
Учасники	Користувач
Передумови	Користувач перейшов на сторінку оптимізації портфелю активів
Постумови	Кількість грошей введена
Основний сценарій	Користувач переходить на сторінку оптимізації і вводить кількість грошей для інвестування в призначене для цього поле

Таблиця 1.10 - Варіант використання UC010

Назва	Перегляд результатів оптимізації в текстовому вікні
Опис	Результати роботи методу оптимізації портфелю відображаються в окремому текстовому вікні
Учасники	Користувач
Передумови	Користувач перейшов на сторінку оптимізації портфелю активів, обрав валюти, що входять до портфелю, обрав період часу для аналізу і суму грошей, яка буде інвестована в портфель

Продовження таблиці 1.10

Постумови	Результат роботи алгоритму оптимізації відображений в текстовому вікні
Основний сценарій	Користувач обрав всі вхідні параметри і натиснув кнопку “Build Optimal Portfolio”

Таблиця 1.11 - Варіант використання UC011

Назва	Повернення до головного меню застосунку
Опис	Користувач може повернутися на головну сторінку застосунку зі сторінки кластерингу чи зі сторінки оптимізації портфелю
Учасники	Користувач
Передумови	Користувач знаходиться на сторінці кластеризації активів або на сторінці оптимізації портфелю активів
Постумови	Користувач знаходиться на головній сторінці застосунку
Основний сценарій	Користувач натискає на кнопку “Main Page”

Функціональні вимоги застосунку описано таблицями 1.12 – 1.19:

Таблиця 1.12 - Опис функціональної вимоги REQ001

Номер	REQ001
Назва	Ведення гіперпараметрів алгоритмів
Опис	Адміністратор може змінювати глобальні налаштування алгоритмів регулярно або за запитом

Таблиця 1.13 - Опис функціональної вимоги REQ002

Номер	REQ002
Назва	Ведення списку алгоритмів кластерингу і оптимізації
Опис	Адміністратор може додавати і видаляти вручну алгоритми кластерингу і методи оптимізації за потреби (розширення функціоналу, виявлення помилок в роботі)

Таблиця 1.14 - Опис функціональної вимоги REQ003

Номер	REQ003
Назва	Вибір довжини історичного періоду для аналізу
Опис	Користувач може вибирати довжину періоду часу, який буде використаний для обчислення параметрів алгоритмів

Таблиця 1.15 - Опис функціональної вимоги REQ004

Номер	REQ004
Назва	Вибір алгоритму кластеризації/ методу оптимізації
Опис	Користувач має можливість обрати алгоритм, що буде використовуватися для кластеризації активів на сторінці кластеризації. Користувач може обрати метод, за допомогою якого буде виконана оптимізація портфелю з обраних активів

Таблиця 1.16 - Опис функціональної вимоги REQ005

Номер	REQ005
Назва	Вибір активів, що входять до портфелю
Опис	Є можливість вибору активів, що будуть входити до портфелю на сторінці оптимізації портфелю активів

Таблиця 1.17 - Опис функціональної вимоги REQ006

Номер	REQ006
Назва	Введення кількості грошей, які будуть інвестовані в портфель
Опис	Користувач може ввести кількість грошей, які будуть інвестовані в портфель в окреме поле на сторінці оптимізації портфелю активів

Таблиця 1.18 - Опис функціональної вимоги REQ007

Номер	REQ007
Назва	Перегляд результатів роботи алгоритму в текстовому вікні
Опис	Результати роботи алгоритмів кластерингу і оптимізацію портфелю відображаються в окремому текстовому вікні

Таблиця 1.19 - Опис функціональної вимоги REQ008

Номер	REQ008
Назва	Повернення до головного меню застосунку
Опис	В застосунку є можливість повернутися на головну сторінку зі сторінки кластерингу чи зі сторінки оптимізації портфелю

Зв'язок між вимогами математичного і програмного забезпечення зображено на рисунку 1.1:

	REQ001 Ведення гіперпараметрів алгоритмів	REQ002 Ведення списку алгоритмів кластеризації і оптимізації	REQ003 Вибір довжини історичного періоду для аналізу	REQ004 Вибір алгоритму кластеризації/ методу оптимізації	REQ005 Вибір активів, що входять до портфелю	REQ006 Введення кількості грошей, які будуть інвестовані в портфель	REQ007 Перегляд результатів роботи алгоритму в текстовому вікні	REQ008 Повернення до головного меню застосунку
UC001 Зміна гіперпараметрів алгоритмів	■							
UC002 Додавання методу оптимізації (алгоритму кластеризації) до списку доступних в застосунку		■						
UC003 Видалення методу оптимізації (алгоритму кластеризації) зі списку доступних в застосунку		■						
UC004 Вибір довжини історичного періоду для аналізу			■					
UC005 Вибір алгоритму кластеризації				■				
UC006 Перегляд результатів кластеризації в текстовому вікні							■	
UC007 Вибір методу оптимізації				■				
UC008 Вибір активів, що входять до портфелю					■			
UC009 Введення кількості грошей, які будуть інвестовані в портфель						■		
UC010 Перегляд результатів оптимізації в текстовому вікні							■	
UC011 Повернення до головного меню застосунку								■

Рисунок 1.1 - Матриця залежності між вимогами і варіантами використання забезпечення

1.4.2 Розроблення нефункціональних вимог

Вимоги до інтерфейсу:

В програмному забезпеченні повинні бути реалізовані такі функції:

- застосунок повинен бути розроблений для використання на комп'ютері;
- необхідна можливість переглянути результат роботи алгоритмів;
- сервіс має використовувати інформацію для аналізу із зовнішніх джерел.

Апаратні та програмні вимоги:

- сервіс повинен запускатися на пристроях з кольоровим дисплеєм;
- сервіс має працювати з операційними системами Ubuntu 16.04, Ubuntu 18.04.

1.5 Постановка цілей та задач дослідження

Призначенням розробки є формування та оптимізація портфелю активів на ринку іноземних валют.

Основними цілями створення математичного і програмного забезпечення є:

- диверсифікація ризиків трейдера;
- зменшення витрат часу трейдера на формування портфелю активів;
- оптимізація торгового портфелю (знаходження оптимального співвідношення між активами).

В рамках даного дослідження мають вирішуватися наступні завдання:

- вибір надійного джерела якісних даних та налагодження ETL процесу;
- здійснення аналізу існуючих рішень та вибір алгоритмів для подальшої реалізації;
- виконання програмної реалізації алгоритмів кластеризації та методів оптимізації;
- підбір оптимальних параметрів алгоритмів;
- дослідження ефективності роботи алгоритмів;
- створення інтерфейсів для отримання результатів роботи алгоритмів.

1.6 Висновки до розділу

В даному розділі:

- проаналізовано відомі в сфері програмні продукти та їх типи ;
- проаналізовано технічні можливості та засоби для реалізації застосунків у даній сфері;
- проведений аналіз вимог до програмного забезпечення;
- сформульовані функціональні вимоги до математичного та програмного забезпечення;
- сформульовані цілі та завдання дослідження.

2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Алгоритми кластеризації

Кластеризацією називають процес розділення множини об'єктів на підмножини за ступенем «схожості» один на одного.

Кластеризація є дуже потужним етапом аналізу даних, тому присутня в багатьох популярних аналітичних рішеннях. Зазвичай аналітики використовують такі алгоритми для розділення всієї вибірки об'єктів на менші групи, а вже потім аналізують кожну з цих груп окремо. Ця стратегія популярна в багатьох сферах діяльності, зокрема в медицині, маркетингу, соціології та інших.

2.1.1 KMeans

Найбільш простий, разом з цим достатньо неточний метод кластеризації в класичній реалізації. Він розбиває множину елементів на підмножини з метою мінімізувати середньоквадратичне відхилення для точок всіх кластерів. Алгоритм є ітеративним, це означає, що під час кожної ітерації відбувається перевизначення центрів мас для кластерів, і перерозподіл об'єктів по нових кластерах відповідно до отриманих значень центрів кластерів. Алгоритм припиняє роботу якщо перевизначення центрів не змінюють конфігурації кластерів.

Переваги:

- висока ефективність при простій реалізації;
- висока якість кластеризації;
- можливість розпаралелення;
- існування великої кількості модифікацій.

Недоліки:

- необхідно завчасно визначати кількість кластерів;

- результат роботи алгоритму залежить від конфігурації початкових кластерів. Класичний варіант передбачає випадковий вибір кластерів, що часто і стає причиною похибки;
- алгоритм не здатен впоратися з задачею, якщо об'єкт можна віднести до кількох кластерів одночасно, або не можна віднести до жодного кластера;
- ітеративний підхід не гарантує знаходження найбільш оптимального розв'язку.
- використання поняття «середнього» (алгоритм не буде працювати з даними, для якого це поняття не існує, наприклад, категорійні дані).

Математичний опис алгоритму.

Вхідні параметри:

- набір з n спостережень, $X = \{x_1, x_2, \dots, x_n\}, x_i \in \mathbb{R}, i = \overline{1, n}$;
- k – кількість кластерів, $k \in \mathbb{N}, k \leq n$

Необхідно розділити множину спостережень X на k кластерів S_1, S_2, \dots, S_k таким чином, що $S_i \cap S_j = \emptyset, i \neq j, \bigcup_{i=1}^k S_i = X$.

Суть роботи:

K-means розділяє набір X на k наборів S_1, S_2, \dots, S_k так, щоб мінімізувати суму квадратів відстаней від кожної точки кластера до центру цього кластера. Якщо $S = S_1, S_2, \dots, S_k$, то дія алгоритму рівносильна пошуку $\operatorname{argmin}_S \sum_{i=1}^k \sum_{x \in S_i} \rho(x, \mu_i)^2$, де μ_i – центри кластерів, $i = \overline{1, k}, \rho(x, \mu_i)$ – функція відстані між x і μ_i .

Кроки алгоритму:

Крок 1. Початковий крок: ініціалізація кластерів. Обирається довільна множина точок $\mu_i, i = \overline{1, k}$, які розглядаються як початкові центри кластерів $\mu_i^{(0)} = \mu_i, i = \overline{1, k}$.

Крок 2. Розподіл векторів по кластерах. Для кожного кроку t : $\forall x_i \in X, i = \overline{1, n}: x_i \in S_i \Leftrightarrow j = \operatorname{argmin}_k \rho(x_i, \mu_k^{(t-1)})^2$.

Крок 3. Перерахунок центрів кластерів. Для кожного кроку t : $\forall i = \overline{1, k}: \mu_i^{(t)} = \frac{1}{|S_i|} \sum_{x \in S_i} x$

Крок 4. Перевірка умови припинення роботи алгоритму.
if $\exists i \in \overline{1, k}: \mu_i^{(t+1)} \neq \mu_i^{(t)}$ *then*
 $t = t + 1$;
 goto 2;
else
 stop [2].

На вхід алгоритму ми подаємо датасет з ануалізованими перформансом і волатильністю для кожної валюти. Як вже було сказано раніше, для використання алгоритму Kmeans ми маємо знати кількість кластерів завчасно. Для того, щоб визначити оптимальне значення цього гіперпараметра використано ліктювий метод (Elbow rule) [3]. Суть методу полягає в тому, що ми послідовно запускаємо кластеринг з різним значенням параметра `n_clusters`, і вважаємо оптимальним те значення, для якого Silhouette score знаходиться в «ліктювому» положенні кривої (тобто значення силуету не зміниться значною мірою, якщо буде змінюватися кількість кластерів). Залежність значення скору силует від кількості кластерів зображено на рисунку 2.1.

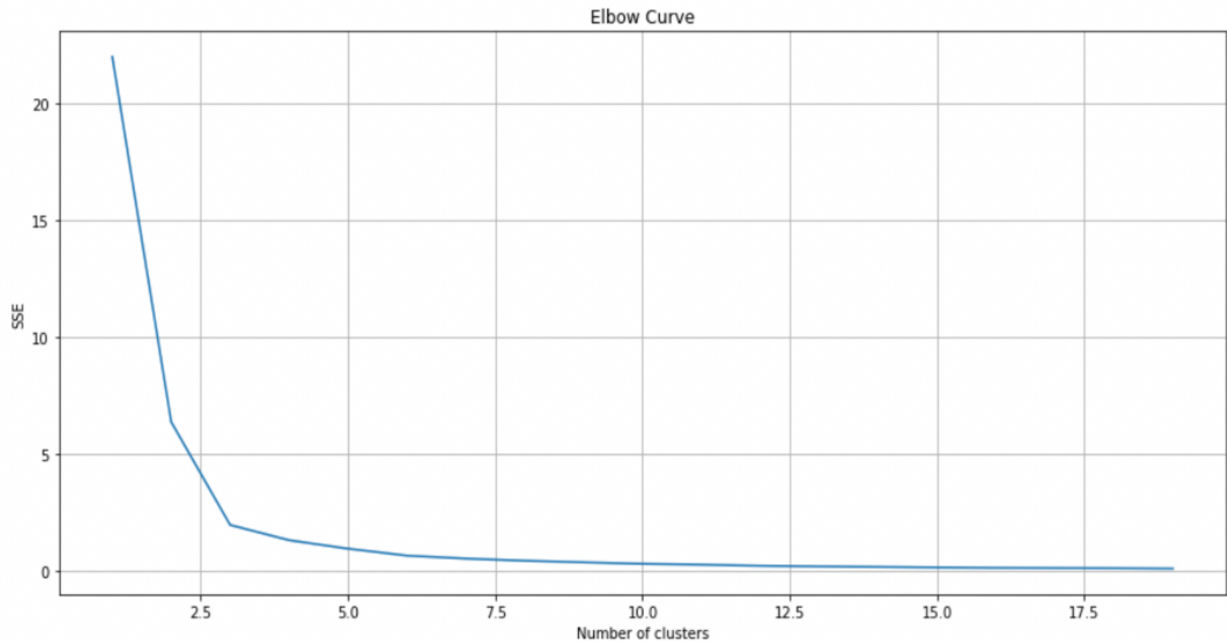


Рисунок 2.1 - Залежність якості кластеризації від кількості кластерів [4]

2.1.2 DBSCAN

DBSCAN (Density-based spatial clustering of applications with noise), як вже можна зрозуміти з назви, оперує щільністю даних. На вхід алгоритму подається матрицю відстані, радіус ϵ -околу і кількість сусідів.

Математичний опис алгоритму

Нехай задана деяка симетрична функція відстані $\rho(x, y)$ і константи ϵ і m .

Крок 1. Назвемо область $E(x)$, для якої $\forall \rho(x, y) \leq \epsilon$, ϵ -окіл об'єкта x .

Крок 2. Кореневим (ядровим) об'єктом степені m називається об'єкт, ϵ -окіл якого містить не менше, ніж m об'єктів: $|E(x)| \geq m$.

Крок 3. Об'єкт p є досяжним (щільно-досяжним) із об'єкта q , якщо $p \in E(q)$ і q – це кореневий об'єкт.

Крок 4. Об'єкт p щільно-досяжний із об'єкта q , якщо $\exists p_1, p_2, \dots, p_n, p_1 = q, p_n = p$, такі, що $\forall i \in \overline{1, n-1}$: p_{i+1} є безпосередньо щільно-досяжним з p_i .

Оберемо будь-який кореневий об'єкт p із датасету і помістимо всіх його безпосередньо щільно-досяжних сусідів в список обходу. Тепер для кожної q із списку позначимо цю точку i , якщо вона також коренева, додамо всіх її сусідів в список обходу. Кластери, які сформовані в результаті такого обходу є максимальними, тобто їх не можна розширити ще однією точкою таким чином, щоб всі умови задовольнялися. Тобто, якщо ми обійшли не всі точки, то можна перезапустити обхід із якого-небудь іншого кореневого об'єкта і новий кластер не поглине попередній.

Головні недоліки алгоритму DBSCAN – це нездатність з'єднувати кластери через проміжки (пропуски) між ними і навпаки, здатність пов'язувати явно різні кластери через велику щільність точок між ними [5].

В дослідженні на вхід алгоритму подається матриця коваріації відносних змін ціни за обраний період. Використовується реалізація з модуля `sklearn.cluster.DBSCAN`, в якій на вхід потрібно подавати ще два параметри: *eps* і *min_samples*. Вони підібрані за допомогою методу пошуку по сітці (Grid Search) [6]. Основна суть цього методу в тому, що ми подаємо на вхід множину можливих значень для кожного гіперпараметра і обираємо функцію оцінки якості кластеризації (в нашому випадку силует), для кожної комбінації параметрів запускаємо кластеризацію і за оптимальні параметри приймаємо ті, для яких функція оцінки якості показала найкращий результат.

Залежність значення силует скору від параметра *eps* зображено на рисунку 2.2, залежність значення силует скору від параметра *min_samples* зображено на рисунку 2.3.

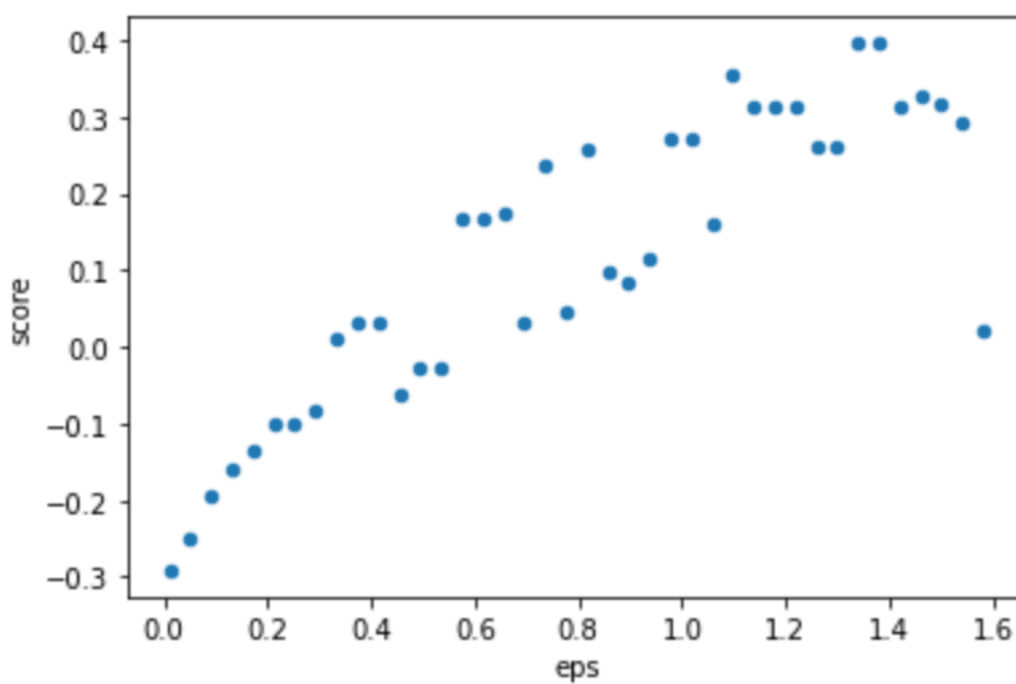


Рисунок 2.2 - Залежність значення силует від параметра *eps*

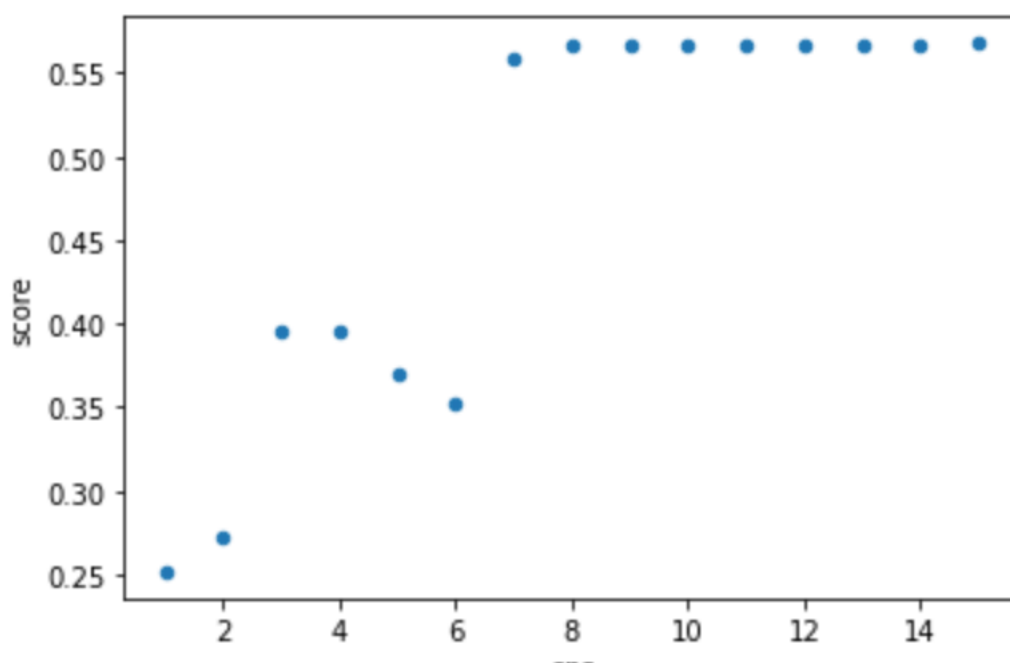


Рисунок 2.3 - Залежність значення силует від параметра *min_samples*

2.2 Алгоритми оптимізації

Портфельна теорія Марковіца (Modern portfolio theory) - це інвестиційна теорія, розроблена Гаррі Марковіцем і опублікована під назвою "Portfolio Selection" у Journal of Finance у 1952 році.

Більш високий ризик пов'язаний з більшою ймовірністю вищого прибутку, а менший ризик із більшою ймовірністю меншого прибутку. Теорія припускає, що інвестори не схильні до ризику, це означає, що з огляду на два портфелі, які пропонують однакову очікувану прибутковість, вони віддадуть перевагу менш ризикованому. Таким чином, інвестор буде брати на себе підвищений ризик лише в тому випадку, якщо це буде компенсовано більшою очікуваною прибутковістю.

Одним із важливих понять, що виникає під час формулювання даної теорії є "диверсифікація". Портфельна теорія Марковіца говорить, що недостатньо розглянути очікуваний ризик і прибутковість одного конкретного активу. Інвестуючи більше ніж в одну акцію, інвестор може зменшити ризикованість свого портфеля.

Задача, яку вирішує дана теорія – це яким чином розподілити свої кошти між елементами портфелю для того, щоб отримати портфель з мінімальним ризиком.

В даному дослідженні представлені різні підходи до реалізації методів даної теорії. Найпростіший з них – метод Монте-Карло [7]. Він полягає в тому, що ми випадково генеруємо велику кількість спостережень і можемо отримати приблизний розв'язок нашої задачі. За умови, що кількість згенерованих спостережень достатньо велика можна стверджувати, що результат має достатньо хорошу точність.

Отже, для того, щоб реалізувати даний метод створений функціонал для генерації довільної кількості випадкових портфелів, а також обрані цільові функції.

Коефіцієнт Шарпа – це показник ефективності активу чи інвестиційного портфеля, який кількісно показує відношення середньої прибутковості до стандартного відхилення активу (портфеля). Чим більший даний показник, тим кращим вважається актив [8].

Волатильність – це статистичний показник, що характеризує інтенсивність зміни ціни [9]. По факту вона є мірою ризику використання певного інструмента за обраний проміжок часу.

Для всіх згенерованих випадкових портфелів обчислюється коефіцієнт Шарпа і волатильність. Найбільш оптимальним відносно коефіцієнта Шарпа можна вважати той портфель, для якого значення показника буде найбільшим, для волатильності – навпаки.

Графічно всі випадково згенеровані випадкові записи і обчислені для них параметри забражені на рисунку 2.4.

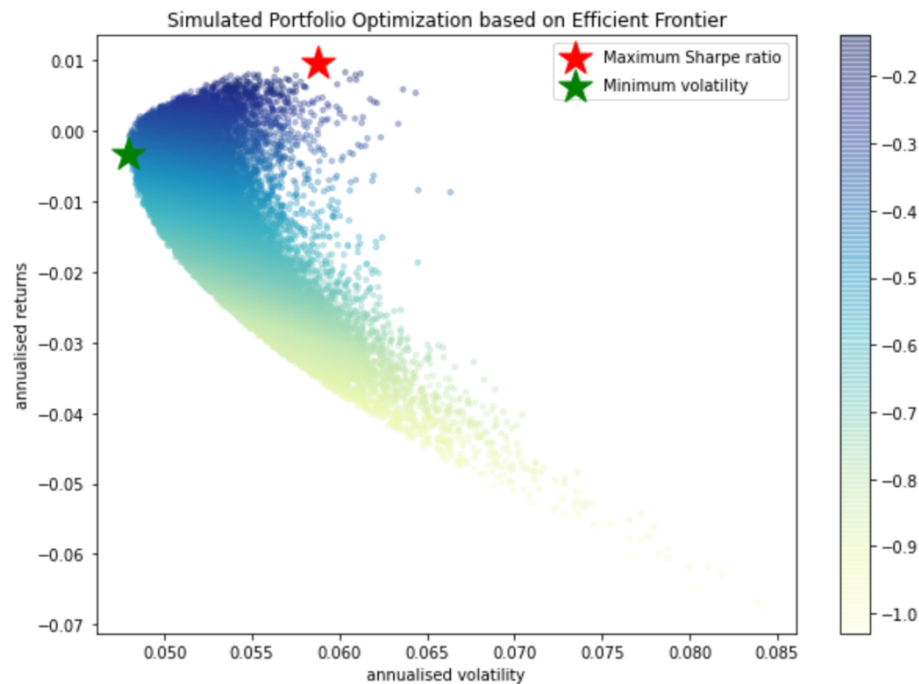


Рисунок 2.4 - Візуалізація роботи методу Монте-Карло

Інший підхід, який представлено в дослідженні – це використання оптимізаційних методів. В даному випадку цільова функція залишається незмінною – коефіцієнт Шарпа або волатильність. Проте замість того, щоб

генерувати велику кількість випадкових розв'язків і з них обирати найкращий, використаний функціонал бібліотеки `scipy.optimize`, а саме метод `minimize` з алгоритмом SLSQP, що вирішує задачу послідовного квадратичного програмування.

Загальний вигляд задачі, яку вирішує даний алгоритм наступний:

$$\begin{aligned} \min_x f(x) \\ \text{subject to: } c_j(x) &= 0, j \in \mathcal{E} \\ c_i(x) &= 0, i \in \mathcal{I} \\ lb_i \leq x_i \leq ub_i, i &= \overline{1, N} \end{aligned}$$

$f(x)$ – обрана цільова функція, на ваги в даному випадку накладаються певні умови: вони мають бути додатні і в сумі давати 1.

Останній із запропонованих методів вирішення проблеми оптимізації – це вирішення все тієї ж задачі квадратичного програмування, проте з використанням спеціального квадратичного солвера (в даному випадку використаний пакет `cvxopt`). Даний спосіб вимагає, щоб всі параметри цільової функції, а саме матриці і вектори (P , q , G , h , A , b) були визначені, тобто на вхід подається не функція, а обраховані завчасно масиви.

Квадратична задача сформульована наступним чином:

$$\begin{aligned} \min \frac{1}{2} x^T P x + q^T x \\ \text{subject to: } Gx \leq h \\ Ax = b \text{ [10].} \end{aligned}$$

Тут P – це матриця коваріації відносних змін цін активів, q – масив з довжиною, що рівна кількості елементів портфелю, G – вектор очікуваних ретьорнів (середні зміни цін для всіх активів), h – цільовий ретьорн (значення, що є цільовим прибутком портфелю), A , b – коефіцієнти для обмеження рівності.

2.3 Висновки до розділу

В даному розділі описано обрані алгоритми кластеризації, процес вибору оптимальних параметрів, наведений опис алгоритмів оптимізації та варіацій їх використання.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання та аналіз програмного забезпечення

Математичне та програмне забезпечення розроблене для Unix подібних операційних систем і справно працює на Mac OS та дистрибутивах Linux (Ubuntu, Mint).

Програмне забезпечення представляє собою десктопний додаток. Для того, щоб спростити розширюваність продукту і надати можливість додавати новий функціонал без ускладнень кодової бази був створений клас WindowController, що ставить у відповідність кожному візуальному розділу (вікну) додатку рядок-ідентифікатор, ініціалізується на початку виконання і в процесі виконання програми для переключання між вікнами необхідно лише змінити ідентифікатор поточного вікна, а для додавання нового візуального елемента необхідно лише додати цей елемент у список відповідностей.

Програмне забезпечення розроблено з використанням мови програмування Python версії 3.6. Так як в межах проекту не передбачається зберігання великих масивів даних (один часовий ряд для одного активу), то для зберігання даних використано Mongo DB. Графічна частина створена за допомогою QtDesigner (застосунок, що дозволяє створити графічне вікно у режимі редактора, а потім згенерувати програмний код, що побудований з використанням відповідних елементів бібліотеки PyQt).

Для обробки і очистки даних використовується модуль pandas, методи машинного навчання (кластеринг) та оптимізації є реалізовані в бібліотеках cvxpy, sklearn. В межах даного дослідження використано готові реалізації алгоритмів з підібраними оптимальними параметрами.

Класи програмного забезпечення описані в таблиці 3.1, їх методи – в таблиці 3.2, інші функції – в таблиці 3.3. Діаграма класів програмного забезпечення наведена в додатках до магістерської дисертації.

3.2 Архітектура програмного забезпечення

Таблиця 3.1 - Опис класів програмного забезпечення

Клас	Опис
Application	Клас головного вікна додатку. Всі візуальні елементи описані в модулі <code>main_window</code>
ClusteringWindow	Вікно кластерингу програмного забезпечення. Візуальні елементи реалізовані в модулі <code>clustering_window</code>
OptimizationWindow	Вікно оптимізації портфелю активів на ринку іноземних валют. Візуальні елементи описані в модулі <code>optimization_window</code>
PandasModel	Клас, призначений для графічного відображення таблиць <code>pandas</code> в текстовому віджеті <code>PyQt</code>
WindowController	Контролер перемикання між вікнами застосунку
FixerioException	Клас, що наслідує базовий <code>Exception</code> і викликається в наслідок виникнення помилки при виклику методів API-інтерфейсу джерела даних
Fixerio	Реалізація обгортки для API-інтерфейсу, що надає джерело даних

Таблиця 3.2 - Опис методів класів програмного забезпечення

Клас	Метод	Опис
ClusteringWindow	__init__	Конструктор класу вікна кластеризації. Ініціалізація візуальних компонент, додавання динамічних вибіркового елементів до списків вибору
ClusteringWindow	switch	Перехід до вказаного вікна застосунку
ClusteringWindow	run_clustering	Запуск роботи методу кластеризації з обраними параметрами
PandasModel	__init__	Конструктор класу. Ініціалізація вхідного датафрейму для відображення
PandasModel	rowCount	Повертає кількість рядків датафрейму
PandasModel	columnCount	Повертає кількість стовпців датафрейму
PandasModel	data	Повертає Qt об'єкт для відображення
OptimizationWindow	__init__	Конструктор класу вікна оптимізації. Ініціалізація візуальних компонент, додавання динамічних вибіркового опцій
OptimizationWindow	switch	Перехід до вказаного вікна застосунку
OptimizationWindow	run_optimization	Запуск роботи методу оптимізації з обраними параметрами
Application	__init__	Конструктор класу головного вікна застосунку. Ініціалізація всіх візуальних компонент і контролера

Продовження таблиці 3.2

Клас	Метод	Опис
Application	switch	Перехід до вказаного вікна застосунку
WindowController	show_window	Відображення вказаного вікна застосунку
WindowController	button_clicked	Метод, що викликається як callback після натискання кнопки для перемикання між вікнами. Закриває попереднє вікно і відображає наступне

Таблиця 3.3 - Опис функцій програмного забезпечення

Функція	Опис
get_data_optimization	Формування набору даних, що буде подаватися на вхід алгоритму оптимізації. На вхід подається список з перерахованими активами, що входять до потенційного портфелю активів, функція повертає ретьорни активів, матрицю коваріації між ними та середні ретьорни для активів

Продовження таблиці 3.3

Функція	Опис
markowitz_portfolio	Обчислення оптимальних вагів портфелю шляхом вирішення задачі квадратичного програмування з використанням засобів бібліотеки cvxopt. На вхід подається матриця коваріації та середні ретърни активів, функція повертає ваги для обраних активів портфелю
portfolio_annualised_performance	Обчислення ануалізованого перфомансу портфелю з заданими вагами. На вхід подаються ваги, середні ретърни і матриця коваріації, функція повертає ануалізований перфоманс та стандартне відхилення активів
random_portfolios	Симуляція заданої кількості випадкових портфелів із заданими параметрами. На вхід подається кількість симуляцій, середні ретърни та матриця коваріацій для активів, функція повертає обчислені цільові метрики для кожної симуляції і симульовані ваги

Продовження таблиці 3.3

Функція	Опис
<code>simulate_ef_with_random_min_vol</code>	Симуляція з метою знаходження портфелю активів з мінімальною волатильністю. На вхід подається матриця коваріації і середні ретьорни, функція повертає ваги для того портфелю, де волатильність найменша
<code>neg_sharpe_ratio</code>	Обчислення коефіцієнта Шарпа з протилежним знаком (це необхідно для того, щоб вирішувати задачу мінімізації, а не максимізації). На вхід подаються ваги, середні ретьорни, матриця коваріації, функція повертає значення коефіцієнта Шарпа (з протилежним знаком)
<code>simulate_ef_with_random_max_sharpe</code>	Симуляція з метою знаходження портфелю активів з максимальним коефіцієнтом Шарпа. На вхід подається матриця коваріації і середні ретьорни, функція повертає ваги для того портфелю, де коефіцієнт Шарпа найбільший
<code>get_valid_currencies</code>	Отримання списку активів, для яких немає викидів в даних для обраного періоду

Продовження таблиці 3.3

Функція	Опис
portfolio_volatility	Обчислення волатильності портфелю. На вхід подаються ваги для портфелю, середні ретьорни і матриця коваріації, функція повертає волатильність портфелю
min_variance	Вирішення задачі мінімізації, де цільова функція – це волатильність портфелю (з використанням засобів модуля scipy). На вхід подаються ваги, середні ретьорни, матриця коваріації, функція повертає оптимальні ваги для портфелю
max_sharpe_ratio	Вирішення задачі мінімізації, де цільова функція – це коефіцієнт Шарпа з протилежним знаком (з використанням засобів модуля scipy). На вхід подаються ваги, середні ретьорни, матриця коваріації, функція повертає оптимальні ваги для портфелю
run_kmeans	Запуск алгоритму kmeans. На вхід подається датафрейм з даними, функція повертає кластери активів

Продовження таблиці 3.3

Функція	Опис
<code>get_data_clustering</code>	Отримання першоджерельних даних для активів, що є в списку валідних. На вхід подається кількість днів для досліджуваного періоду, функція повертає датафрейм з даними
<code>get_data_clustering_kmeans</code>	Формування вхідного набору даних для алгоритму <code>kmeans</code> . На вхід подається таймсерія з цінами активів, функція повертає набір даних, де кожному активу відповідає один запис
<code>get_data_clustering_dbscan</code>	Формування вхідного набору даних для алгоритму <code>dbscan</code> . На вхід подається таймсерія з цінами активів, функція повертає набір даних для роботи <code>dbscan</code>
<code>run_dbscan</code>	Запуск алгоритму <code>dbscan</code> . На вхід подається датафрейм з даними, функція повертає кластери активів

3.3 Аналіз безпеки даних

Програмне забезпечення отримує інформацію, що вважається відкритою, так як її може отримати будь-хто зареєстрований на сайті джерела даних. На даний момент не передбачається необхідності реєстрації користувача в застосунку, тому ризику втрати персональних даних також немає.

Однією з гарантій того, що дані отримані під час безпечного з'єднання з джерелом даних є використання HTTPS протоколу – він вважається

криптографічно сек'юрним, тобто достатньо захищеним, так як має додаткове шифрування/автентифікацію між HTTPS і TCP.

3.4 Керівництво користувача

Після запуску додатку користувач знаходиться на головній сторінці застосунку (рисунок 3.1).

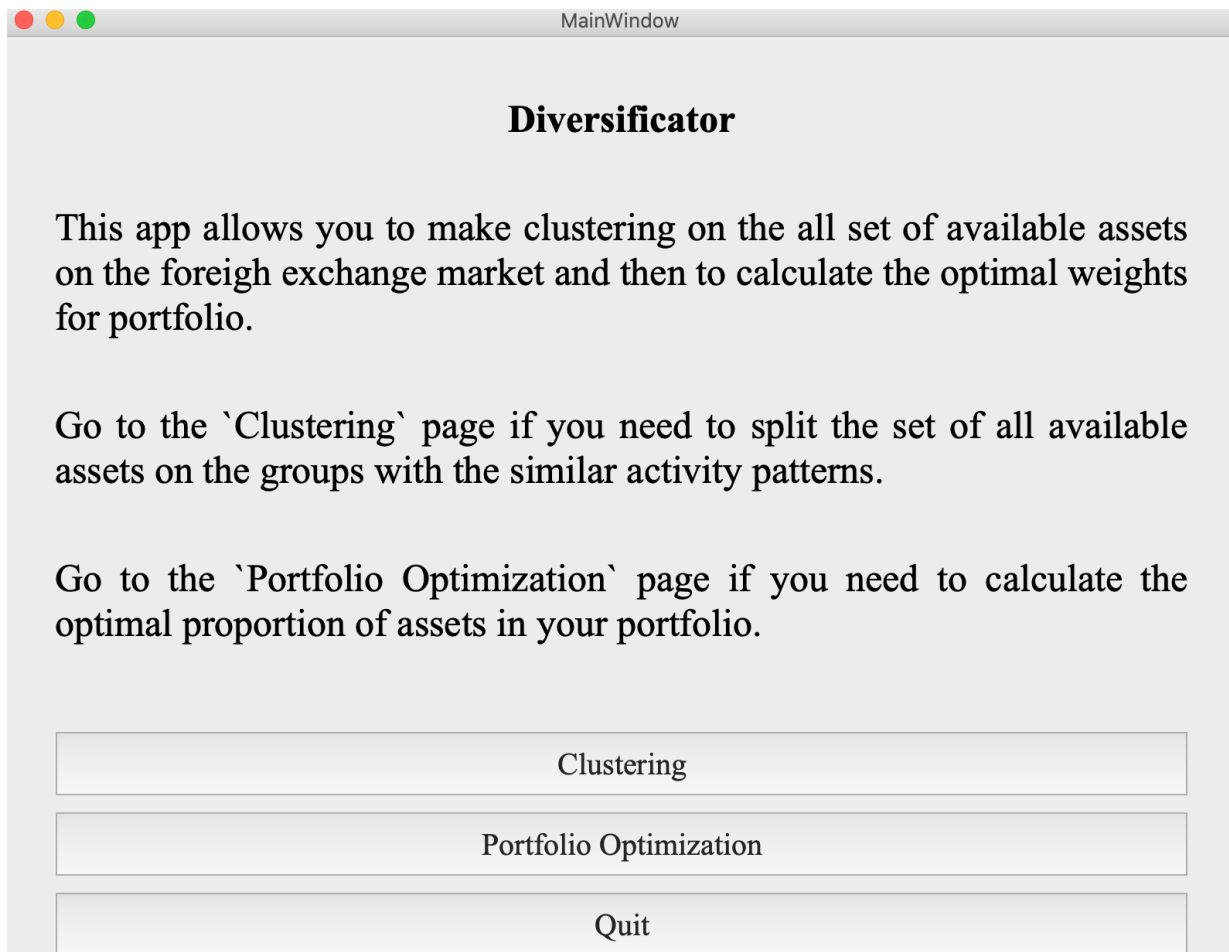


Рисунок 3.1 – Головна сторінка застосунку

Для переходу до вікна кластерингу необхідно натиснути на кнопку з написом 'Clustering'.

Після переходу на вікно кластеризації (рисунок 3.2) необхідно обрати алгоритм, тривалість історичного періоду для аналізу і натиснути кнопку 'Run Clustering'. Результат роботи алгоритму відобразиться в текстовому вікні справа.

Clustering Page

This page allows you to perform clustering and find groups of assets with the similar patterns in their behaviour.

First of all you need to choose clustering method in the first box, then you need to choose the period length which will be used for analysis (default values is `KMeans` and 365 days correspondingly). Then press **Run Clustering** and you will see the result on the right-side text browser.

KMeans

365

Run Clustering

Cluster 1: AED, AFN, ALL, AMD, ANG, AUD, AWG, AZN, BAM, BBD, BDT, BGN, BHD, BIF, BMD, BND, BOB, BSD, BTN, BWP, BYR, BZD, CAD, CHF, CNY, CRC, CUC, CUP, CZK, DJF, DKK, DZD, EGP, ERN, EUR, FJD, FKP, GBP, GGP, GIP, GMD, GNF, GTQ, GYD, HKD, HNL, HRK, HUF, IDR, ILS, IMP, INR, IQD, ISK, JEP, JMD, JOD, JPY, KES, KHR, KMF, KPW, KRW, KWD, KYD, KZT, LAK, LBP, LKR, LRD, LTL, LVL, LYD, MAD, MDL, MKD, MMK, MNT, MOP, MRO, MUR, MVR, MWK, MYR, NIO, NOK, NPR, NZD, OMR, PAB, PEN, PGK, PHP, PKR, PLN, PYG, QAR, RON, RSD, RWF, SAR, SBD, SEK, SGD, SLL, SOS, SRD, STD, SVC, SYP, THB, TJS, TMT, TOP, TTD, TWD, TZS, UGX, USD, UZS, VEF, VND, VUV, WST, XAF, XAU, XCD, XDR, XOF, XPF, YER, ZMK, ZWL

Cluster 2: CVE, IRR

Cluster 3: AOA, ARS, BRL, BYN, CDF, CLF, CLP, COP, DOP, ETB, GEL, HTG, KGS, LSL, MXN, MZN, NAD, NGN, RUB, SCR, SDG, SZL, TRY, UAH, UYU, ZAR, ZMW

Cluster 4: BTC, MGA

Cluster 5: SHP, XAG

Main Menu

Рисунок 3.2 – Вікно кластерингу активів

Для повернення до головного меню необхідно натиснути кнопку `Main Menu`. Користувач знову опиняється в головному меню застосунку (рисунок 3.1).

Для переходу до сторінки оптимізації необхідно натиснути кнопку `Portfolio Optimization`.

На вікні оптимізації (рисунок 3.3) необхідно обрати валюти, які будуть входити до портфелю активів, алгоритм оптимізації, тривалість історичної інформації для аналізу і ввести кількість грошей, які будуть інвестовані в торговий портфель. Після цього необхідно натиснути кнопку `Build Optimal Portfolio`. Результат буде відображений в текстовому вікні справа. Перша колонка таблиці – валюта, друга – відсоток капіталу, що потрібно вкласти в

даний актив, третя колонка – кількість грошей, що необхідно вкласти в цей актив (з розрахунку на суму, що ввів користувач).

Optimization Page

This page allows you to perform calculate optimal weights for assets in portfolio.

First of all you need to choose optimization method in the first box and period length which will be used for analysis (default values is 'Markowitz portfolio' and 365 days correspondingly). Also you need to input total amount of money you want to invest.

Then press 'Build optimal portfolio' and you will see the result on the right-side text browser.

KWD
 KYD
 KZT
 LAK
 LBP
 LKR
 LRD
 LSL
 LTL

Markowitz Portfolio

365

100000,00

Build optimal portfolio

	1	2	3
1 AUD	10.008%	10007.99999...	
2 CAD	75.68%	75680.00000...	
3 XAU	0.267%	267.0	
4 RUB	14.045%	14045.0	

Main Menu

Рисунок 3.3 – Вікно оптимізації

3.5 Висновки до розділу

В розділі 3 проведено аналіз програмного забезпечення, описано класи та методи програмного забезпечення, архітектуру і обрані засоби для реалізації в цілому.

4 РЕЗУЛЬТАТИ ЕКСПЕРЕМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

4.1 Мета та порядок досліджень

Для дослідження ефективності застосунку обрано 8 валют, які входять до досліджуваного портфелю.

Необхідно дослідити ефективність роботи алгоритмів та їх швидкодію.

4.2 Дослідження ефективності алгоритмів оптимізації

До портфелю входять наступні валюти: AUD – австралійський долар, BTC – біткоїн, CAD – канадський долар, INR – індійська рупія, JPY – японська єна, NZD – новозеландський долар, UAH – українська гривня, XAU – золото.

За допомогою розв’язання задачі квадратичного програмування отримано наступне оптимальне співвідношення між активами:

Таблиця 4.1 – Оптимальне співвідношення між активами портфелю

Актив	AUD	BTC	CAD	INR	JPY	NZD	UAH	XAU
Вага	0.40105	0.00514	0.11535	0.0082	0.3126	0.00105	0.145	0.01158

Оптимальний портфель буде порівнюватися з так званим Equal-weights portfolio – портфелем, в якому ваги активів, що входять до його складу однакові, тобто, в даному випадку рівні 0.125. Для того, щоб порівняти два портфелі обчислені наступні показники:

- ануалізована волатильність – є мірою розкиду ретърнів портфелю, інакше кажучи, волатильність – це міра ризикованості, чим більше волатильність, тим більш ризикованим є портфель [11];

- коефіцієнт Шарпа – показник «ефективності» портфелю, чим вищий коефіцієнт Шарпа, тим кращим є актив чи портфель, для якого він розрахований;

– очікуваний ретьорн – середнє значення для ретьорнів за обраний часовий проміжок, тобто чим більший цей показник, тим більш прибутковим в середньому є актив [12].

Результати наведені в таблиці 4.2:

Таблиця 4.2 - Порівняння ефективності рівнозваженого та оптимального портфелів

	Ануалізована волатильність	Коефіцієнт Шарпа	Ануалізований очікуваний ретьорн
Рівнозважений портфель	6.28%	-0.2562	-1.609%
Оптимальний портфель	3.35%	1.9485	6.527%

З отриманих результатів видно, що оптимальний портфель є вдвічі менш ризикованим і значно ефективнішим, так як має дуже високий коефіцієнт Шарпа. Натомість, якби користувач створив рівнозважений портфель, то з великою імовірністю він втратив би гроші, так як очікуваний ретьорн такого портфелю негативний. Також про високу неефективність свідчить і негативний коефіцієнт Шарпа (за класичним визначенням, навіть додатній, проте близький до нуля коефіцієнт Шарпа вже є ознакою низької ефективності активу) [13].

4.3 Дослідження швидкодії алгоритмів оптимізації

В розробленому математичному та програмному забезпеченні реалізовано 5 різних способів отримання оптимального портфелю:

– перебір з вибором найменш волатильного зразка (Efficient Frontier with random generation (min Volatility)), суть якого полягає в тому, що спочатку генерується велика кількість випадкових портфелів, для кожного з них обчислюється волатильність і результатом роботи є той, для якого волатильність найменша;

- перебір з вибором зразка з найкращим коефіцієнтом Шарпа (Efficient Frontier with random generation (max Sharpe Ratio)), суть якого полягає в тому, що спочатку генерується велика кількість випадкових портфелів, для кожного з них обчислюється коефіцієнт Шарпа і результатом роботи є той, для якого коефіцієнт є найбільшим;
- розв’язання задачі оптимізації (мінімізація), де цільовою функцією є волатильність ('Efficient Frontier with optimization (min Volatility));
- розв’язання задачі оптимізації (мінімізація), де цільовою функцією є коефіцієнт Шарпа (точніше негативний коефіцієнт Шарпа – це необхідно для того, щоб перейти від задачі максимізації до задачі мінімізації) (Efficient Frontier with optimization (max Sharpe Ratio));
- розв’язання задачі квадратичного програмування (Markowitz Portfolio).

Для визначення середнього часу, витраченого на обчислення використана бібліотека мови Python time, результати наведено в таблиці 4.3

Таблиця 4.3 - Середній час, необхідний для роботи алгоритму

Метод оптимізації	Час, витрачений на обчислення (в секундах)
Efficient Frontier with random generation (min Volatility)	9.04583
Efficient Frontier with random generation (max Sharpe Ratio)	9.45769
Efficient Frontier with optimization (min Volatility)	0.05036
Efficient Frontier with optimization (max Sharpe Ratio)	0.01562
Markowitz Portfolio	0.14124

Способи обчислення вагів методом генерації багатьох випадкових зразків є найповільнішими, адже вимагають майже 10 секунд, що є доволі багато. До того ж, ці методи не гарантують отримання оптимального результату, так як найбільш оптимальна точка може бути не згенерована і не потрапить до досліджуваної вибірки, проте за умови генерації доволі великої кількості зразків можна вважати, що буде отриманий достатньо оптимальний розв'язок. Ці методи є неефективними, так як використовують генератори випадкових чисел. В перспективі використання більш оптимального генератора може пришвидшити роботу методів, проте це не відкидає їх неефективності.

Разом з цим бачимо, що всі інші розроблені методи показують хорошу швидкодію, так як для отримання результату потрібні долі секунди, тобто це задовільний результат і дані алгоритми придатні для використання в застосунках.

4.4 Висновки до розділу

В розділі 4 доведено ефективність використання методу оптимізації, а також визначено середній час, що необхідний на роботу всіх розроблених методів, проаналізовано їх швидкодію. На основі отриманих результатів можна стверджувати, що алгоритми дають оптимальний результат і не потребують багато часу для його отримання.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

5.1 Опис основної ідеї проекту

Таблиця 5.1 - Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Формування та оптимізація торгового портфелю активів на ринку іноземних валют	Виявлення груп валют на фореکсі, що мають подібні патерни в поведінці	Зменшення витрат часу на вибір активів, які будуть входити до портфелю
	Оптимізація торгового портфелю (знаходження оптимального співвідношення активів)	Зменшення витрат часу на розподіл грошей між активами портфелю
	Диверсифікація ризиків трейдера	Одноразове зменшення ризикованості торгового портфелю і збільшення його імовірного прибутку

Таблиця 5.2 - Визначення сильних, слабких та нейтральних характеристик

	Потенційні товари / концепції конкурентів		
	Мій проект	ForexAnalytix	Autochartist
W (слабка сторона)	<ul style="list-style-type: none"> На даний момент існує лише десктопний додаток Використання даних денної частоти 	<ul style="list-style-type: none"> Відсутність безкоштовної версії продукту Відсутність автоматизованих засобів диверсифікації 	<ul style="list-style-type: none"> Відсутність автоматизованих засобів диверсифікації

Продовження таблиці 5.2

N (нейтральна сторона)	– Відсутність аналітичних показників та візуалізацій	– Наявність додатків для різних платформ – Наявність навчального матеріалу	– Наявність додатків для різних платформ – Тісна співпраця з медіа ресурсами
S (сильна сторона)	– Можливість інтегруватися з будь-яким іншим веб-сервісом, за умови, якщо є можливість надсилати та отримувати GET/POST запити – Низькі апаратні вимоги – Простий у використанні	– Аналіз багатьох джерел інформації – Надання підтримки користувачам в режимі реального часу – Інтуїтивно зрозумілий інтерфейс – Наявність інфографіки та аналітичних показників – Швидкодія	– Наявність безкоштовної пробної підписки з обмеженим функціоналом – Аналіз різноманітних джерел інформації – Наявність інфографіки та аналітичних показників

5.2 Технологічний аудит ідеї проекту

Таблиця 5.3 – Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Створення власної API обгортки для отримання даних зі стороннього сервісу	Модуль requests	Наявна	Доступна
2.	Збереження історичних даних у власному сховищі	MongoDB	Наявна	Доступна
3.	Кластеризація активів для виділення груп зі схожою поведінкою	Модуль sklearn	Наявна	Доступна
4.	Оптимізація портфелю обраних активів	Модулі cvxpy, cvxopt, scipy	Наявні	Доступні
5.	Можливість перегляду результатів роботи алгоритму у візуальному інтерфейсі	Qt (PyQt)	Наявна	Доступна
<p>Висновок: для реалізації проекту наявні і доступні всі необхідні технології, тому його реалізувати можливо.</p> <p>Обрані технології реалізації проекту: мова програмування – Python, база даних – MongoDB, фреймворки алгоритмів машинного навчання та оптимізації – sklearn, cvxpy, візуальний інтерфейс – Qt.</p>				

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.4 – Попередня характеристика ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	500000 доларів (з розрахунку на 500 проданих річних підписок)
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Формування торгового портфелю, знаходження оптимального співвідношення елементів в ньому	Досвідчені трейдери, трейдери-початківці, крупні торгові організації, аналітики	Крупні компанії, що торгують грошима клієнтів більш вимогливі до точності роботи алгоритмів	Інтуїтивна зрозумілість інтерфейсу, мінімальна необхідна кількість дій для отримання результату

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість продуктом (відсутність попиту)	Вузькоспрямованість продукту, відсутність розуміння проблеми у потенційних клієнтів	Проведення більш спеціалізованої рекламної кампанії, донесення до клієнтів необхідності використання забезпечення, проведення онлайн-занять з демонстрацією використання розробленого забезпечення
2	Невелика кількість функціоналу	Кількість доступних функцій та алгоритмів невелика	Розробка і додавання нових функцій, за умови збереження стабільності системи, так як функціонал повний, наголошення на простоті використання продукту клієнтом

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розвиток сфери ринку іноземних валют	Збільшення конкуренції, підвищення волатильності основних інструментів	Більш активна пропаганда (реклама) свого математичного та програмного забезпечення

Продовження таблиці 5.7

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
2	Зростання кількості трейдерів початківців	Ринок переповнений новачками, які програють гроші з великою ймовірністю	Надання послуг цій цільовій аудиторії з метою досягнення ними позитивного досвіду при торгівлі і тим самим збереження своїх клієнтів
3	Поява нових алгоритмів і технологій	Сфера машинного навчання і аналізу даних розвивається, тому з'являються нові дослідження та алгоритми	Додавання до стеку математичного і програмного забезпечення нових алгоритмів
4	Велика кількість відкритих даних	Зростання кількості сервісів, що надають можливість безкоштовно (або дуже дешево) отримати дані для дослідження і роботи програмного забезпечення	Розробка нових алгоритмів і покращення вже готових з новими даними, роль посередника (постачання даних клієнтам, яким це потрібно)

Продовження таблиці 5.7

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
5	Відсутність повноцінного продукту- альтернативи	Наявні продукти не дають можливості отримати такий же результат настільки швидко і за допомогою одного й того ж програмного продукту	Забезпечення стабільності роботи існуючого функціоналу, додавання нового функціоналу

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – олігополія	На ринку доступні додатки та сервіси зі схожим функціоналом, проте повноцінного конкурента немає	Перш за все забезпечити стабільність і якість роботи базових алгоритмів і запустити додаток для перших клієнтів, розвивати існуючий функціонал
2. За рівнем конкурентної боротьби – глобальний	Світовий ринок не має повноцінного конкурента	Пропозиція розробленого програмного забезпечення вже існуючій аудиторії
3. За галузевою ознакою – внутрішньогалузева	Розробка призначена лише для використання на ринку форекс	Підтримка і покращення існуючих функцій

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
4. Конкуренція за видами товарів – товарно-видова	Призначення розробки – формування та оптимізація торгового портфелю Відмінність – наявність всього необхідного функціоналу в одному місці	Якість та маркетинг програмного забезпечення
5. За характером конкурентних переваг – нецінова	Для користувачів важливо отримувати якісні послуги швидко та з мінімальною витратою зусиль	Супроводження існуючого функціоналу і взаємодія з користувачами (надання зворотного зв'язку)
6. За інтенсивністю – марочна	Для того, щоб продукт став популярним необхідна хороша репутація	Надання якісного функціоналу, який не можуть надати конкуренти, спілкування і взаємодія з користувачами

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Autochartlist	Forex Analytix	Відсутні	Кількість споживачів, доступність інформації	Більша популярність через захвачені сегменти ринку
Висновки	В прямого конкурента є своя аудиторія і попит	Потенційний конкурент надає якісні послуги, але не є повним аналогом	-	Якість програмного забезпечення і зацікавленість аудиторії в ньому	Обмеження роботи через товари заміники

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкість роботи алгоритмів	Клієнти не змушені довго чекати на результат роботи алгоритму (робота займає 2-3 секунди)
2	Висока якість програмного забезпечення	ПЗ має зручний та інтуїтивно зрозумілий інтерфейс, працює безперебійно

Продовження таблиці 5.10

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
3	Відповідність потребам цільової аудиторії	Програмне забезпечення швидко виконує поставлену перед ним задачу і отриманий результат повністю покриває потреби користувача

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін ПЗ

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним ПЗ						
			-3	-2	-1	0	+1	+2	+3
1	Швидкість роботи алгоритмів	18			+				
2	Висока якість програмного забезпечення	15		+					
3	Відповідність потребам цільової аудиторії	20	+						

Таблиця 5.12 – SWOT-аналіз стартап проєкту

Сильні сторони: <ul style="list-style-type: none"> – Зручність у використанні – можливість отримати кінцевий результат в межах одного застосунку 	Слабкі сторони: <ul style="list-style-type: none"> – Наявність менш якісних аналогів, які задовольняють користувачів на перший погляд
---	---

Продовження таблиці 5.12

Можливості:	Загрози:
<ul style="list-style-type: none"> – Ріст і розвиток ринку іноземних валют – Збільшення кількості джерел даних – Ріст цільової аудиторії 	<ul style="list-style-type: none"> – Наявність великої кількості конкурентів, що пропонують «гарну обкладинку» – Неготовність клієнтів платити

Таблиця 5.13 – Альтернативи ринкового впровадження стартап проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів ринкової поведінки)	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка веб-інтерфейсу застосунку	Висока	1 місяць
2	Створення API-інтерфейсу для передачі результатів роботи алгоритмів стороннім компаніям	Середня	2 тижні

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	Трейдери	Готові	Високий	Висока	Просто

Продовження таблиці 5.14

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів прийняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
2	Крупні торгові компанії	Потребують переговорів	Середній	Висока	Складно
3	Аналітики, що працюють самі на себе	Готові	Середній	Середня	Просто
Обрано цільові групи: трейдери, аналітики.					

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Орієнтація поточної моделі на трейдерів (досвідчених і початківців)	Стратегія концентрованого маркетингу	Трейдери, яким потрібно формування торгового портфелю	Стратегія спеціалізації
2	Орієнтація поточної моделі на аналітиків	Поступове захоплення ринку, акцент на якість, а не на швидкість	Обрана цільова група, що принесе максимум прибутку (ті аналітики, яким потрібні додаткові інструменти)	Стратегія спеціалізації

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є стартап «першопрохідцем» на ринку?	Чи буде дана компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде дана компанія копіювати основні характеристики товару конкурентів і які?	Стратегія конкурентної поведінки
1	Ні	Пошук нових користувачів	Ні. Основний акцент під час просування даного забезпечення – простота у використанні. Зайві ускладнення будуть недоречні	Стратегія займання конкурентної ніші

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару від цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Швидкість роботи, мінімальна кількість витрачених сил і часу для отримання результату	Стратегія спеціалізації	Трейдерам необхідний засіб для швидкого формування торгового портфелю, аналітикам необхідний додатковий засіб в арсенал вже наявних для надання більш якісних послуг	Доступність, швидкість, унікальність, простота

5.5 Розроблення маркетингової програми проекту

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі, або такі, що потрібно створити)
1	Спрощення процесу формування торгового портфелю для пересічного користувача	Необхідність натискання трьох кнопок для отримання повної інформації про склад портфелю	Більшість конкурентів мають надто складний інтерфейс, тому для отримання результат треба забагато часу

Продовження таблиці 5.18

2	Поєднання формування та оптимізації в одному додатку	В одному додатку є можливість здійснити кластеризацію, і одразу ж знайти оптимальне співвідношення між обраними активами	Конкуренти не поєднують даний функціонал в межах одного застосунку
---	--	--	--

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Математичне та програмне забезпечення для формування та оптимізації торгового портфелю ринку іноземних валют		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Кластеризація	1. М	1. Тх
	2. Оптимізація	2. М	2. Тх
	3. Дешевизна	3. М	3. Тл
	4. Швидкість роботи	4. М	4. Е
	Якість: математичне та програмне забезпечення		
Марка: НТУУ КПІ ім. Ігоря Сікорського			
III. Товар із підкріпленням	Потенційний користувач може дізнатися про можливості застосування даного забезпечення з рекламних та навчальних роликів, після продажу завжди є можливість звернутися в службу підтримки чи залишити зворотній зв’язок на сайті		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Таблиця 5.20 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Надання безкоштовної (частково обмеженої) та повної версій продукту	Простота в отриманні доступу до програмного забезпечення, оплаті послуг	Середня	Безпосередня

Таблиця 5.21 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, що використовують цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Орієнтація на якісний і простий у використан ні продукт	Спеціалізовані блоги, сайти компаній	Простота у використанні, швидкість отримання результату	Показати, що створення портфелю займає лічені хвилини навіть для недосвідченого трейдера свій портфель	«Почни торгувати вже зараз», «Розумний торговий портфель – твій портфель»

5.6 Висновки до розділу

В даному розділі проведено аналіз ринку, на якому буде розвиватися розроблений стартап-проект, вивчено переваги та недоліки проекту в

порівнянні з конкурентами та аналогами. Також розглянуто та передбачено ризики і можливі варіанти розвитку даного проекту.

Дане програмне забезпечення може розвиватися в будь-якому іншому вигляді, наприклад, веб-застосунок, чи просто окремий API інтерфейс, так як комп'ютерний застосунок – це лише метод відображення результатів. Тому за необхідності даний проект дуже легко пристосувати до нових вимог і передати результати будь-якому клієнту.

ВИСНОВКИ

В даній магістерській роботі було розроблено математичне та програмне забезпечення оптимізації портфелю активів на ринку іноземних валют. Програма написана мовою Python, використані бібліотеки pandas, numpy, sklearn, cvxopt, pymongo, pyrestful та інші. Інформація зберігається в MongoDB.

В першому розділі проведено аналіз вимог до програмного забезпечення, описано основні сценарії роботи програми та проведено аналіз конкурентів.

В другому розділі описано математичне забезпечення магістерської роботи, а саме алгоритми кластеризації та алгоритми оптимізації, набори даних, що подаються на вхід, вибір оптимальних гіперпараметрів.

В третьому розділі описано етап моделювання програмного забезпечення, описано основні класи та функції застосунку, розроблене керівництво користувача.

Четвертий розділ містить опис експериментального дослідження, в якому порівняно модернізований алгоритм оптимізації зі звичайним перебором великої кількості випадкових спостережень. За результатами досліджень розроблений алгоритм відпрацював за 0.01 секунду, натомість базовий варіант потребує майже 10 секунд для отримання результату, що є абсолютно неефективно.

Наукова новизна полягає у модифікації алгоритму оптимізації, а саме покращенні його швидкодії.

Усі поставлені задачі виконані і головна мета досягнута – користувач швидко і без зайвих дій зі свого боку отримує результати роботи алгоритмів.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Electronic trading platform [Електронний ресурс]: (Стаття) / Algowiki – Електрон. дан. (1 файл). – 2020. – Режим доступу: https://en.wikipedia.org/wiki/Electronic_trading_platform.
- 2) Алгоритм k средних (k-means) [Електронний ресурс]: (Стаття) / Algowiki – Електрон. дан. (1 файл). – 2018. – Режим доступу: [https://algowiki-project.org/ru/Алгоритм_k_средних_\(k-means\)](https://algowiki-project.org/ru/Алгоритм_k_средних_(k-means)).
- 3) Elbow method (clustering) [Електронний ресурс]: (Стаття) / Wikipedia – Електрон. дан. (1 файл). – 2020. – Режим доступу: [https://en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering)).
- 4) Ясенова А.В., Халус О.А. Застосування алгоритмів кластеризації на ринку іноземних валют // V Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020» Матеріали конференції. – 2020. – С.
- 5) Интересные алгоритмы кластеризации, часть вторая: DBSCAN [Електронний ресурс]: (Стаття) / Habr – Електрон. дан. (1 файл). – 2017. – Режим доступу: <https://habr.com/ru/post/322034/>.
- 6) Hyperparameter optimization [Електронний ресурс]: (Стаття) / Wikipedia – Електрон. дан. (1 файл). – 2019. – Режим доступу: https://en.wikipedia.org/wiki/Hyperparameter_optimization#Grid_search.
- 7) Метод Монте-Карло [Електронний ресурс]: (Стаття) / Wikipedia – Електрон. дан. (1 файл). – 2020. – Режим доступу: https://uk.wikipedia.org/wiki/Метод_Монте-Карло.
- 8) Коэффициент Шарпа [Електронний ресурс]: (Стаття) / Economic definition – Електрон. дан. (1 файл). – 2020. – Режим доступу: https://economic-definition.com/Other_branches_of_mathematics/Koefficient_Sharpa_Sharpe_Ratio_eto.html.

9) Волатильність [Електронний ресурс]: (Стаття) / НБУ – Електрон. дан. (1 файл). – 2004. – Режим доступу: https://old.bank.gov.ua/control/uk/publish/article?art_id=123193&cat_id=122978.

10) Quadratic Programming with Python and CVXOPT [Електронний ресурс]: (Стаття) / MIT – Електрон. дан. (1 файл). – 2018. – Режим доступу: <https://courses.csail.mit.edu/6.867/wiki/images/a/a7/Qp-cvxopt.pdf>.

11) Calculating annualized volatility of stock returns [Електронний ресурс]: (Стаття) / Quant – Електрон. дан. (1 файл). – 2020. – Режим доступу: <https://quant.stackexchange.com/questions/30519/calculating-annualized-volatility-of-stock-returns>.

12) Definition of "Rate of Return" in Forex Trading [Електронний ресурс]: (Стаття) / tradingcharts – Електрон. дан. (1 файл). – 2020. – Режим доступу: <https://forex.tradingcharts.com/glossary/Forex+Trading/Rate+of+Return.html>.

13) The Sharpe Ratio Broke Investors' Brains [Електронний ресурс]: (Стаття) / Institutional Investor – Електрон. дан. (1 файл). – 2020. – Режим доступу: <https://www.institutionalinvestor.com/article/b1p62z599ns4pd/The-Sharpe-Ratio-Broke-Investors-Brains>

ДОДАТОК А ТЕКСТ ПРОГРАМНОГО КОДУ

Тексти програмного коду

Програмне та математичне забезпечення оптимізації портфелю активів на ринку іноземних валют

(Найменування програми (документа))

DVD-R

(Вид носія даних)

13 арк, 2994 Кб

(Обсяг програми, арк., Кб)

API обгортка для отримання даних з джерела

```

from __future__ import unicode_literals
import datetime

try:
    from urllib.parse import urljoin
except ImportError: # For Python 2
    from urlparse import urljoin # noqa

import requests

BASE_URL = 'http://data.fixer.io/api/'

LATEST_PATH = 'latest'

class FixerioException(BaseException):
    pass

class Fixerio(object):

    def __init__(self, access_key, symbols=None):
        self.access_key = access_key
        self.symbols = symbols

    def _create_payload(self, symbols):
        payload = {'access_key': self.access_key}
        if symbols is not None:
            payload['symbols'] = ','.join(symbols)

        return payload

    def latest(self, symbols=None):
        try:
            symbols = symbols or self.symbols
            payload = self._create_payload(symbols)

            url = BASE_URL + LATEST_PATH

            response = requests.get(url, params=payload)

```

```

        response.raise_for_status()

        return response.json()
    except requests.exceptions.RequestException as ex:
        raise FixerioException(str(ex))

def historical_rates(self, date, symbols=None):
    try:
        if isinstance(date, datetime.date):
            date = date.isoformat()

        symbols = symbols or self.symbols
        payload = self._create_payload(symbols)

        url = BASE_URL + date

        response = requests.get(url, params=payload)

        response.raise_for_status()

        return response.json()
    except requests.exceptions.RequestException as ex:
        raise FixerioException(str(ex))

```

Автоматично згенерований код для вікна кластеризації

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

```

class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(800, 600)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(Form.sizePolicy().hasHeightForWidth())
        Form.setSizePolicy(sizePolicy)
        Form.setMinimumSize(QtCore.QSize(800, 600))
        Form.setMaximumSize(QtCore.QSize(800, 600))
        self.verticalLayoutWidget = QtWidgets.QWidget(Form)

```

```

self.verticalLayoutWidget.setGeometry(QtCore.QRect(0, 0, 801, 601))
self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
self.vertical_layout_1 = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
self.vertical_layout_1.setContentsMargins(0, 0, 0, 0)
self.vertical_layout_1.setObjectName("vertical_layout_1")
self.label_clustering_title = QtWidgets.QLabel(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setFamily("Times New Roman")
self.label_clustering_title.setFont(font)
self.label_clustering_title.setObjectName("label_clustering_title")
self.vertical_layout_1.addWidget(self.label_clustering_title)
self.horizontal_layout_1_1 = QtWidgets.QHBoxLayout()
self.horizontal_layout_1_1.setObjectName("horizontal_layout_1_1")
self.vertical_layout_1_1_1 = QtWidgets.QVBoxLayout()
self.vertical_layout_1_1_1.setObjectName("vertical_layout_1_1_1")
self.label_clustering_description = QtWidgets.QLabel(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(15)
font.setStrikeOut(False)
font.setStyleStrategy(QtGui.QFont.PreferDefault)
self.label_clustering_description.setFont(font)
self.label_clustering_description.setTextFormat(QtCore.Qt.AutoText)
self.label_clustering_description.setWordWrap(True)
self.label_clustering_description.setObjectName("label_clustering_description")
self.vertical_layout_1_1_1.addWidget(self.label_clustering_description)
self.combo_box_method_selection =
QtWidgets.QComboBox(self.verticalLayoutWidget)
self.combo_box_method_selection.setMinimumSize(QtCore.QSize(0, 50))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(16)
self.combo_box_method_selection.setFont(font)
self.combo_box_method_selection.setObjectName("combo_box_method_selection")
self.vertical_layout_1_1_1.addWidget(self.combo_box_method_selection)
self.spin_box_period_duration = QtWidgets.QSpinBox(self.verticalLayoutWidget)
self.spin_box_period_duration.setMinimumSize(QtCore.QSize(0, 50))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(15)

```

```

self.spin_box_period_duration.setFont(font)
self.spin_box_period_duration.setMinimum(90)
self.spin_box_period_duration.setMaximum(730)
self.spin_box_period_duration.setProperty("value", 365)
self.spin_box_period_duration.setObjectName("spin_box_period_duration")
self.vertical_layout_1_1_1.addWidget(self.spin_box_period_duration)
self.button_run_clustering = QtWidgets.QPushButton(self.verticalLayoutWidget)
self.button_run_clustering.setMinimumSize(QtCore.QSize(0, 50))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(20)
self.button_run_clustering.setFont(font)
self.button_run_clustering.setObjectName("button_run_clustering")
self.vertical_layout_1_1_1.addWidget(self.button_run_clustering)
self.horizontal_layout_1_1.addLayout(self.vertical_layout_1_1_1)
self.text_browser_clustering_results =
QtWidgets.QTextBrowser(self.verticalLayoutWidget)
self.text_browser_clustering_results.setMinimumSize(QtCore.QSize(300, 401))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
self.text_browser_clustering_results.setFont(font)

self.text_browser_clustering_results.setObjectName("text_browser_clustering_results")
self.horizontal_layout_1_1.addWidget(self.text_browser_clustering_results)
self.vertical_layout_1.addLayout(self.horizontal_layout_1_1)
self.horizontal_layout_1_2 = QtWidgets.QHBoxLayout()
self.horizontal_layout_1_2.setObjectName("horizontal_layout_1_2")
spacerItem = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.horizontal_layout_1_2.addItem(spacerItem)
self.button_back_to_main_menu =
QtWidgets.QPushButton(self.verticalLayoutWidget)
self.button_back_to_main_menu.setMinimumSize(QtCore.QSize(200, 50))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(20)
self.button_back_to_main_menu.setFont(font)
self.button_back_to_main_menu.setObjectName("button_back_to_main_menu")
self.horizontal_layout_1_2.addWidget(self.button_back_to_main_menu)

```

```

self.vertical_layout_1.addLayout(self.horizontal_layout_1_2)

self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)

def retranslateUi(self, Form):
    _translate = QtCore.QCoreApplication.translate
    Form.setWindowTitle(_translate("Form", "Form"))
    self.label_clustering_title.setText(_translate("Form",
    "<html><head/><body><p align=\\\"center\\\"><span style=\\\" font-size:36pt; font-weight:600;\\\">Clustering
Page</span></p></body></html>"))
    self.label_clustering_description.setText(_translate("Form",
    "<html><head/><body><p align=\\\"justify\\\"><span style=\\\" font-size:18pt;\\\">This page
allows you to perform clustering and find groups of assets with the similar patterns in
their behaviour. </span><br/></p><p align=\\\"justify\\\"><span style=\\\" font-
size:18pt;\\\">First of all you need to choose clustering method in the first box, then
you need to choose the period length which will be used for analysis (default values is
`KMeans` and 365 days correspondingly). Then press `</span><span style=\\\" font-
size:18pt; font-weight:600;\\\">Run Clustering</span><span style=\\\" font-size:18pt;\\\">`
and you will see the result on the right-side text browser.</span></p></body></html>"))
    self.button_run_clustering.setText(_translate("Form", "Run Clustering"))
    self.button_back_to_main_menu.setText(_translate("Form", "Main Menu"))

```

Автоматично згенерований код для вікна оптимізації

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

```

class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(800, 600)
        Form.setMinimumSize(QtCore.QSize(800, 600))
        Form.setMaximumSize(QtCore.QSize(800, 600))
        self.verticalLayoutWidget = QtWidgets.QWidget(Form)
        self.verticalLayoutWidget.setGeometry(QtCore.QRect(6, 3, 791, 601))
        self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
        self.vertical_layout_1 = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.vertical_layout_1.setContentsMargins(0, 0, 0, 0)
        self.vertical_layout_1.setObjectName("vertical_layout_1")
        self.label_optimization_title = QtWidgets.QLabel(self.verticalLayoutWidget)
        font = QtGui.QFont()

```

```

font.setFamily("Times New Roman")
self.label_optimization_title.setFont(font)
self.label_optimization_title.setObjectName("label_optimization_title")
self.vertical_layout_1.addWidget(self.label_optimization_title)
self.horizontal_layout_1_1 = QtWidgets.QHBoxLayout()
self.horizontal_layout_1_1.setObjectName("horizontal_layout_1_1")
self.vertical_layout_1_1_1 = QtWidgets.QVBoxLayout()
self.vertical_layout_1_1_1.setObjectName("vertical_layout_1_1_1")
self.label_optimization_description                                     =
QtWidgets.QLabel(self.verticalLayoutWidget)
    font = QtGui.QFont()
    font.setFamily("Times New Roman")
    font.setPointSize(16)
    self.label_optimization_description.setFont(font)
    self.label_optimization_description.setWordWrap(True)

self.label_optimization_description.setObjectName("label_optimization_description")
    self.vertical_layout_1_1_1.addWidget(self.label_optimization_description)
    self.list_widget_currencies_selection                               =
QtWidgets.QListWidget(self.verticalLayoutWidget)
    font = QtGui.QFont()
    font.setFamily("Times New Roman")
    font.setPointSize(14)
    self.list_widget_currencies_selection.setFont(font)

self.list_widget_currencies_selection.setSelectionMode(QtWidgets.QAbstractItemView.MultiSelection)

self.list_widget_currencies_selection.setObjectName("list_widget_currencies_selection")
    self.vertical_layout_1_1_1.addWidget(self.list_widget_currencies_selection)
    self.combo_box_method_selection                                   =
QtWidgets.QComboBox(self.verticalLayoutWidget)
    self.combo_box_method_selection.setMinimumSize(QtCore.QSize(0, 40))
    font = QtGui.QFont()
    font.setFamily("Times New Roman")
    font.setPointSize(16)
    self.combo_box_method_selection.setFont(font)
    self.combo_box_method_selection.setObjectName("combo_box_method_selection")
    self.vertical_layout_1_1_1.addWidget(self.combo_box_method_selection)
    self.spin_box_period_duration = QtWidgets.QSpinBox(self.verticalLayoutWidget)

```



```

self.spin_box_period_duration.setMinimumSize(QtCore.QSize(0, 30))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(16)
self.spin_box_period_duration.setFont(font)
self.spin_box_period_duration.setMinimum(90)
self.spin_box_period_duration.setMaximum(730)
self.spin_box_period_duration.setProperty("value", 365)
self.spin_box_period_duration.setObjectName("spin_box_period_duration")
self.vertical_layout_1_1_1.addWidget(self.spin_box_period_duration)

```

```

self.spin_double_spin_money = QtWidgets.QDoubleSpinBox(self.verticalLayoutWidget)
self.spin_double_spin_money.setMinimumSize(QtCore.QSize(0, 30))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(16)
self.spin_double_spin_money.setFont(font)
self.spin_double_spin_money.setMinimum(0)
self.spin_double_spin_money.setMaximum(1000000000)
self.spin_double_spin_money.setProperty("value", 100000)
self.spin_double_spin_money.setObjectName("spin_double_spin_money")
self.vertical_layout_1_1_1.addWidget(self.spin_double_spin_money)

```

```

self.button_run_optimization = QtWidgets.QPushButton(self.verticalLayoutWidget)
self.button_run_optimization.setMinimumSize(QtCore.QSize(0, 40))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(20)
self.button_run_optimization.setFont(font)
self.button_run_optimization.setObjectName("button_run_optimization")
self.vertical_layout_1_1_1.addWidget(self.button_run_optimization)
self.horizontal_layout_1_1.addLayout(self.vertical_layout_1_1_1)
self.tableView = QtWidgets.QTableView(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(14)
self.tableView.setFont(font)
self.tableView.setObjectName("tableView")

```

```

self.horizontal_layout_1_1.addWidget(self.tableView)
self.vertical_layout_1.addLayout(self.horizontal_layout_1_1)
self.horizontal_layout_1_2 = QtWidgets.QHBoxLayout()
self.horizontal_layout_1_2.setObjectName("horizontal_layout_1_2")
spacerItem = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Minimum)
self.horizontal_layout_1_2.addItem(spacerItem)
self.button_back_to_main_menu =
QtWidgets.QPushButton(self.verticalLayoutWidget)
self.button_back_to_main_menu.setMinimumSize(QtCore.QSize(200, 50))
font = QtGui.QFont()
font.setFamily("Times New Roman")
font.setPointSize(20)
self.button_back_to_main_menu.setFont(font)
self.button_back_to_main_menu.setObjectName("button_back_to_main_menu")
self.horizontal_layout_1_2.addWidget(self.button_back_to_main_menu)
self.vertical_layout_1.addLayout(self.horizontal_layout_1_2)

self.retranslateUi(Form)
QtCore.QMetaObject.connectSlotsByName(Form)

def retranslateUi(self, Form):
    _translate = QtCore.QCoreApplication.translate
    Form.setWindowTitle(_translate("Form", "Form"))
    self.label_optimization_title.setText(_translate("Form", "<html><head/><body><p
align=\"center\"><span style=\" font-size:24pt; font-weight:600;\">Optimization
Page</span></p></body></html>"))
    self.label_optimization_description.setText(_translate("Form",
"<html><head/><body><p align=\"justify\"><span style=\" font-size:14pt;\">This page
allows you to perform calculate optimal weights for assets in portfolio. </span></p><p
align=\"justify\"><span style=\" font-size:14pt;\">First of all you need to choose
optimization method in the first box and period length which will be used for analysis
(default values is `</span><span style=\" font-size:14pt; font-weight:600;\">Marcowitz
portfolio</span><span style=\" font-size:14pt;\">` and </span><span style=\" font-
size:14pt; font-weight:600;\">365 days</span><span style=\" font-size:14pt;\">
correspondingly). Also you need to input total amount of money you want to
invest.</span></p><p align=\"justify\"><span style=\" font-size:14pt;\">Then press
`</span><span style=\" font-size:14pt; font-weight:600;\">Build optimal
portfolio</span><span style=\" font-size:14pt;\">` and you will see the result on the
right-side text browser.</span></p></body></html>"))

```

```

        self.button_run_optimization.setText(_translate("Form", "Build optimal
portfolio"))
        self.button_back_to_main_menu.setText(_translate("Form", "Main Menu"))

```

Модуль оптимізації

```

import numpy as np
import pandas as pd
import cvxopt as opt
import cvxopt.solvers as optsolvers
import scipy.optimize as sco

def get_data_optimization(assets, n_days):
    df = pd.read_csv('loaded_rates.csv')
    df = df.iloc[::-1].tail(n_days)
    df['date'] = pd.to_datetime(df['date'])
    returns = df.set_index('date')[assets].pct_change().dropna()

    avg_rets = returns.mean()
    cov_mat = returns.cov()

    return returns, cov_mat, avg_rets

def markowitz_portfolio(cov_mat, exp_rets):
    target_ret = exp_rets.quantile(0.7)
    n = len(cov_mat)

    P = opt.matrix(cov_mat.values)
    q = opt.matrix(0.0, (n, 1))
    G = opt.matrix(np.vstack((-exp_rets.values, -np.identity(n))))
    h = opt.matrix(np.vstack((-target_ret, + np.zeros((n, 1)))))
    A = opt.matrix(1.0, (1, n))
    b = opt.matrix(1.0)

    optsolvers.options['show_progress'] = False
    sol = optsolvers.qp(P, q, G, h, A, b)

    if sol['status'] != 'optimal':
        return "Convergence problem"

```

```

else:
    weights = pd.Series(sol['x'], index=cov_mat.index)
    return weights

def portfolio_annualised_performance(weights, mean_returns, cov_matrix):
    returns = np.sum(mean_returns * weights) * 252
    std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights))) * np.sqrt(252)
    return std, returns

def random_portfolios(num_portfolios, mean_returns, cov_matrix, risk_free_rate):
    results = np.zeros((3, num_portfolios))
    weights_record = []
    for i in range(num_portfolios):
        weights = np.random.random(len(mean_returns))
        weights /= np.sum(weights)
        weights_record.append(weights)
        portfolio_std_dev, portfolio_return = portfolio_annualised_performance(weights,
mean_returns, cov_matrix)
        results[0, i] = portfolio_std_dev
        results[1, i] = portfolio_return
        results[2, i] = (portfolio_return - risk_free_rate) / portfolio_std_dev
    return results, weights_record

def simulate_ef_with_random_max_sharpe(cov_mat, exp_rets):
    num_portfolios = 25000
    risk_free_rate = 0.0178
    results, weights = random_portfolios(num_portfolios, exp_rets, cov_mat,
risk_free_rate)
    max_sharpe_idx = np.argmax(results[2])
    sdp, rp = results[0, max_sharpe_idx], results[1, max_sharpe_idx]
    max_sharpe_allocation = pd.Series(weights[max_sharpe_idx], index=cov_mat.columns,
name='values')
    return max_sharpe_allocation

def simulate_ef_with_random_min_vol(cov_mat, exp_rets):
    num_portfolios = 25000

```

```

risk_free_rate = 0.0178
results, weights = random_portfolios(num_portfolios, exp_rets, cov_mat,
risk_free_rate)
min_vol_idx = np.argmin(results[0])
sdp_min, rp_min = results[0, min_vol_idx], results[1, min_vol_idx]
min_vol_allocation = pd.Series(weights[min_vol_idx], index=cov_mat.columns,
name='values')
return min_vol_allocation

def neg_sharpe_ratio(weights, mean_returns, cov_matrix, risk_free_rate):
    p_var, p_ret = portfolio_annualised_performance(weights, mean_returns, cov_matrix)
    return -(p_ret - risk_free_rate) / p_var

def max_sharpe_ratio(mean_returns, cov_matrix, risk_free_rate):
    num_assets = len(mean_returns)
    args = (mean_returns, cov_matrix, risk_free_rate)
    constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
    bound = (0.0, 1.0)
    bounds = tuple(bound for asset in range(num_assets))
    result = sco.minimize(neg_sharpe_ratio, num_assets*[1./num_assets,], args=args,
                          method='SLSQP', bounds=bounds, constraints=constraints)
    return result

def portfolio_volatility(weights, mean_returns, cov_matrix):
    return portfolio_annualised_performance(weights, mean_returns, cov_matrix)[0]

def min_variance(mean_returns, cov_matrix):
    num_assets = len(mean_returns)
    args = (mean_returns, cov_matrix)
    constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
    bound = (0.0, 1.0)
    bounds = tuple(bound for asset in range(num_assets))

    result = sco.minimize(portfolio_volatility, num_assets*[1./num_assets,], args=args,
                          method='SLSQP', bounds=bounds, constraints=constraints)

    return result

```

```

def ef_with_selected_max_sharpe(cov_mat, exp_rets):
    risk_free_rate = 0.0178
    max_sharpe = max_sharpe_ratio(exp_rets, cov_mat, risk_free_rate)
    sdp, rp = portfolio_annualised_performance(max_sharpe['x'], exp_rets, cov_mat)
    max_sharpe_allocation = pd.Series(max_sharpe.x, index=cov_mat.columns,
name='values')
    return max_sharpe_allocation

def ef_with_selected_min_vol(cov_mat, exp_rets):
    risk_free_rate = 0.0178

    min_vol = min_variance(exp_rets, cov_mat)
    sdp_min, rp_min = portfolio_annualised_performance(min_vol['x'], exp_rets, cov_mat)
    min_vol_allocation = pd.Series(min_vol.x, index=cov_mat.columns, name='values')
    return min_vol_allocation

optimization_methods = {
    'Markowitz Portfolio': markowitz_portfolio,
    'Efficient Frontier with random generation (max Sharpe Ratio)':
simulate_ef_with_random_max_sharpe,
    'Efficient Frontier with random generation (min Volatility)':
simulate_ef_with_random_min_vol,
    'Efficient Frontier with optimization (max Sharpe Ratio)':
ef_with_selected_max_sharpe,
    'Efficient Frontier with optimization (min Volatility)': ef_with_selected_min_vol
}

```

Модуль кластеризації

```

import pandas as pd
import numpy as np
from sklearn.cluster import KMeans

def get_valid_currencies(num_days=100):
    df = pd.read_csv('loaded_rates.csv')
    df = df.iloc[:::-1].tail(num_days).drop('date', axis=1)
    annualized_return = df.pct_change().mean() * 252
    return [c for c in df.columns if c not in
list(annualized_return[annualized_return.abs() > 1].index)]

```

```

def get_data_clustering(n_days):
    assets = get_valid_currencies(n_days)
    df = pd.read_csv('loaded_rates.csv')
    df = df.iloc[:::-1].tail(n_days)[['date'] + assets]
    return df

def get_data_clustering_kmeans(df):
    df = df.drop('date', axis=1)
    # Calculating the annual Returns for each currency pairs in our dataset
    Calc = df.pct_change().mean() * 252
    Calc = pd.DataFrame(Calc)
    Calc.columns = ['Returns']
    # Calculating the Volatility (standard deviation) for each currency pairs in our
dataset
    Calc['Volatility'] = df.pct_change().std() * (252 ** .5)
    Calc_Conv = np.asarray([np.asarray(Calc['Returns']),
                             np.asarray(Calc['Volatility'])]).T
    return Calc_Conv, Calc.index

def get_data_clustering_dbscan(df):
    df = df.set_index('date')
    returns = df.pct_change().dropna()
    iv = returns.T
    iv.head()
    variation = iv[:]
    X = variation.copy().T

    X /= X.std(axis=0)
    X = X.T.dropna().T
    return X

from sklearn.metrics import silhouette_samples, silhouette_score
def run_kmeans(df):
    Calc_Conv, ccy_labels = get_data_clustering_kmeans(df)
    range_of_k = list(range(5, 12))
    Mean_Silhouette_Scores = []
    for k in range_of_k:
        km = KMeans(n_clusters=k)

```

```

    labels = km.fit_predict(Calc_Conv)
    Silhouette_Smpls = silhouette_samples(Calc_Conv, labels)
    Avg_Silhouette_Score = np.mean(Silhouette_Smpls)
    Mean_Silhouette_Scores.append(Avg_Silhouette_Score)

Optimum_Clusters =
range_of_k[Mean_Silhouette_Scores.index(max(Mean_Silhouette_Scores))]

km = KMeans(n_clusters=Optimum_Clusters)
labels = km.fit_predict(Calc_Conv)
Cluster_details = [(name, cluster) for name, cluster in zip(ccy_labels, labels)]
clusters = []
for i in range(0, Optimum_Clusters):
    tmp = []
    for detail in Cluster_details:
        if detail[1] == i:
            tmp.append(detail[0])
    clusters.append(tmp)
res = ''
for i in range(len(clusters)):
    res += '<p><b>Cluster %i:</b> %s</p>' % ((i + 1), ', '.join(clusters[i]))
    # res += 'Cluster ' + str(i + 1) + ' = ' + str(clusters[i]) + '\n'
return res

from sklearn import cluster
def run_dbscan(df):
    X = get_data_clustering_dbscan(df)
    model = cluster.DBSCAN(eps=1.14, min_samples=2)
    model.fit(X.cov())
    names = np.array(list(X.columns))
    labels = np.array(model.labels_)
    n_labels = model.labels_.max()

    res = ''
    for i in range(n_labels + 1):
        res += '<p><b>Cluster %i:</b> %s</p>' % ((i + 1), ', '.join(names[labels ==
i]))
    return res

clustering_methods = {
    'KMeans': run_kmeans,

```



```

        'DBSCAN': run_dbscan
    }

```

Основний виконуваний файл

```

import sys
from PyQt5 import QtWidgets, QtCore
import main_window, optimization_window, clustering_window
import pandas as pd

def get_valid_currencies(num_days=100):
    df = pd.read_csv('loaded_rates.csv')
    df = df.iloc[::-1].tail(num_days).drop('date', axis=1)
    annualized_return = df.pct_change().mean() * 252
    return [c for c in df.columns if c not in
list(annualized_return[annualized_return.abs() > 1].index)]

# clustering_methods = ['Markowitz Portfolio', 'Efficient Frontier']
from clustering_module import clustering_methods, get_data_clustering
class ClusteringWindow(QtWidgets.QWidget, clustering_window.Ui_Form):

    switch_window = QtCore.pyqtSignal(str)

    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.button_back_to_main_menu.clicked.connect(lambda: self.switch('main'))
        self.button_run_clustering.clicked.connect(self.run_clustering)

        self.combo_box_method_selection.addItem(clustering_methods.keys())

    def switch(self, str_key):
        self.switch_window.emit(str_key)

    def run_clustering(self):
        method_name = str(self.combo_box_method_selection.currentText())
        n_days = self.spin_box_period_duration.value()

        data = get_data_clustering(n_days)
        res = clustering_methods[method_name](data)
        self.text_browser_clustering_results.setText(res)

```

```
from optimization_module import optimization_methods, get_data_optimization
```

```
class PandasModel(QtCore.QAbstractTableModel):
```

```
    def __init__(self, data, parent=None):
        QtCore.QAbstractTableModel.__init__(self, parent)
        self._data = data
```

```
    def rowCount(self, parent=None):
        return len(self._data.values)
```

```
    def columnCount(self, parent=None):
        return self._data.columns.size
```

```
    def data(self, index, role=QtCore.Qt.DisplayRole):
        if index.isValid():
            if role == QtCore.Qt.DisplayRole:
                return QtCore.QVariant(str(
                    self._data.values[index.row()][index.column()]
                ))
        return QtCore.QVariant()
```

```
class OptimizationWindow(QtWidgets.QWidget, optimization_window.Ui_Form):
```

```
    switch_window = QtCore.pyqtSignal(str)
```

```
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.button_back_to_main_menu.clicked.connect(lambda: self.switch('main'))
        self.button_run_optimization.clicked.connect(self.run_optimization)

        self.list_widget_currencies_selection.addItems(get_valid_currencies(300))
        self.combo_box_method_selection.addItems(list(optimization_methods.keys()))
```

```
    def switch(self, str_key):
        self.switch_window.emit(str_key)
```

```

def run_optimization(self):
    method_name = str(self.combo_box_method_selection.currentText())
    currencies = [str(item.text()) for item in
self.list_widget_currencies_selection.selectedItems()]
    n_days = self.spin_box_period_duration.value()
    money = self.spin_double_spin_money.value()

    returns, cov_mat, avg_rets = get_data_optimization(currencies, n_days)
    res = optimization_methods[method_name](cov_mat, avg_rets)
    tmp = pd.DataFrame({'currency': res.index, 'weight': res.values})
    tmp['weight'] = round(100*tmp['weight'], 3)
    tmp['$ amount'] = round(tmp['weight'] * money / 100, 3)
    tmp['weight'] = tmp['weight'].apply(lambda x: str(x) + '%')
    m = PandasModel(tmp)

    self.tableView.setModel(m)
    self.tableView.show()

```

```

class Application(QtWidgets.QMainWindow, main_window.Ui_MainWindow):

```

```

    switch_window = QtCore.pyqtSignal(str)

```

```

    def __init__(self):

```

```

        super().__init__()
        self.setupUi(self)
        self.button_clustering.clicked.connect(lambda: self.switch('clustering'))
        self.button_optimization.clicked.connect(lambda: self.switch('optimization'))
        self.button_quit.clicked.connect(lambda: self.close())

```

```

    def switch(self, str_key):
        self.switch_window.emit(str_key)

```

```

class WindowController:

```

```

    bnt_window = {
        'clustering': ClusteringWindow,
        'optimization': OptimizationWindow,

```

```

        'main': Application
    }

    def __init__(self):
        pass

    def show_window(self, str_key):
        self.window = self.bnt_window[str_key]()
        self.window.show()
        self.window.switch_window.connect(self.button_clicked)

    def button_clicked(self, str_key):
        self.window.close()
        self.show_window(str_key)

def main():
    app = QtWidgets.QApplication(sys.argv)
    controller = WindowController()
    controller.show_window('main')
    sys.exit(app.exec_())

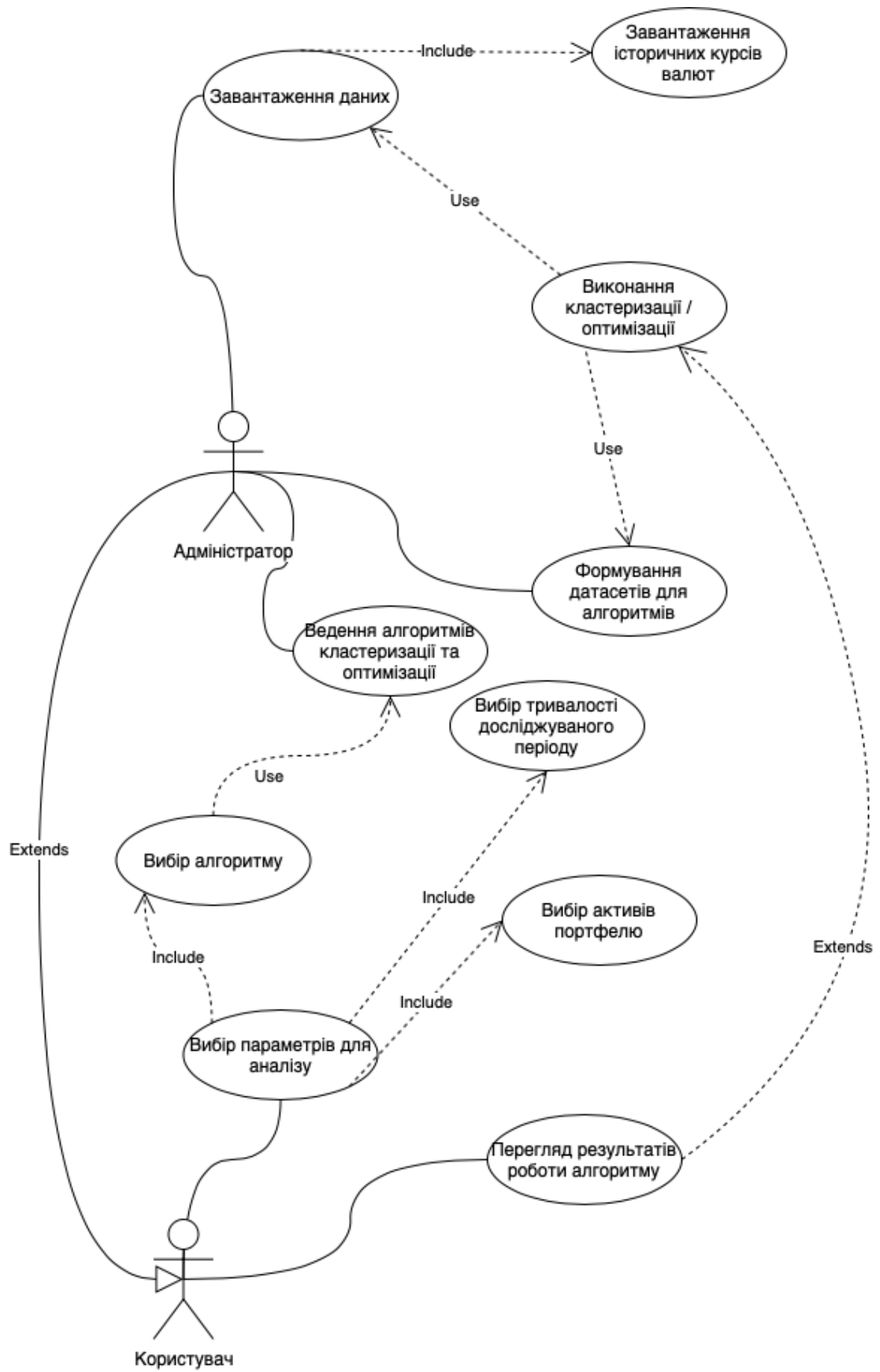
if __name__ == '__main__':
    main()

```

ДОДАТОК Б СХЕМА СТРУКТУРНА ВАРІАНТІВ ВИКОРИСТАННЯ

***Схема структурна варіантів використання
Програмне та математичне забезпечення оптимізації
портфелю активів на ринку іноземних валют***

(Найменування програми (документа))

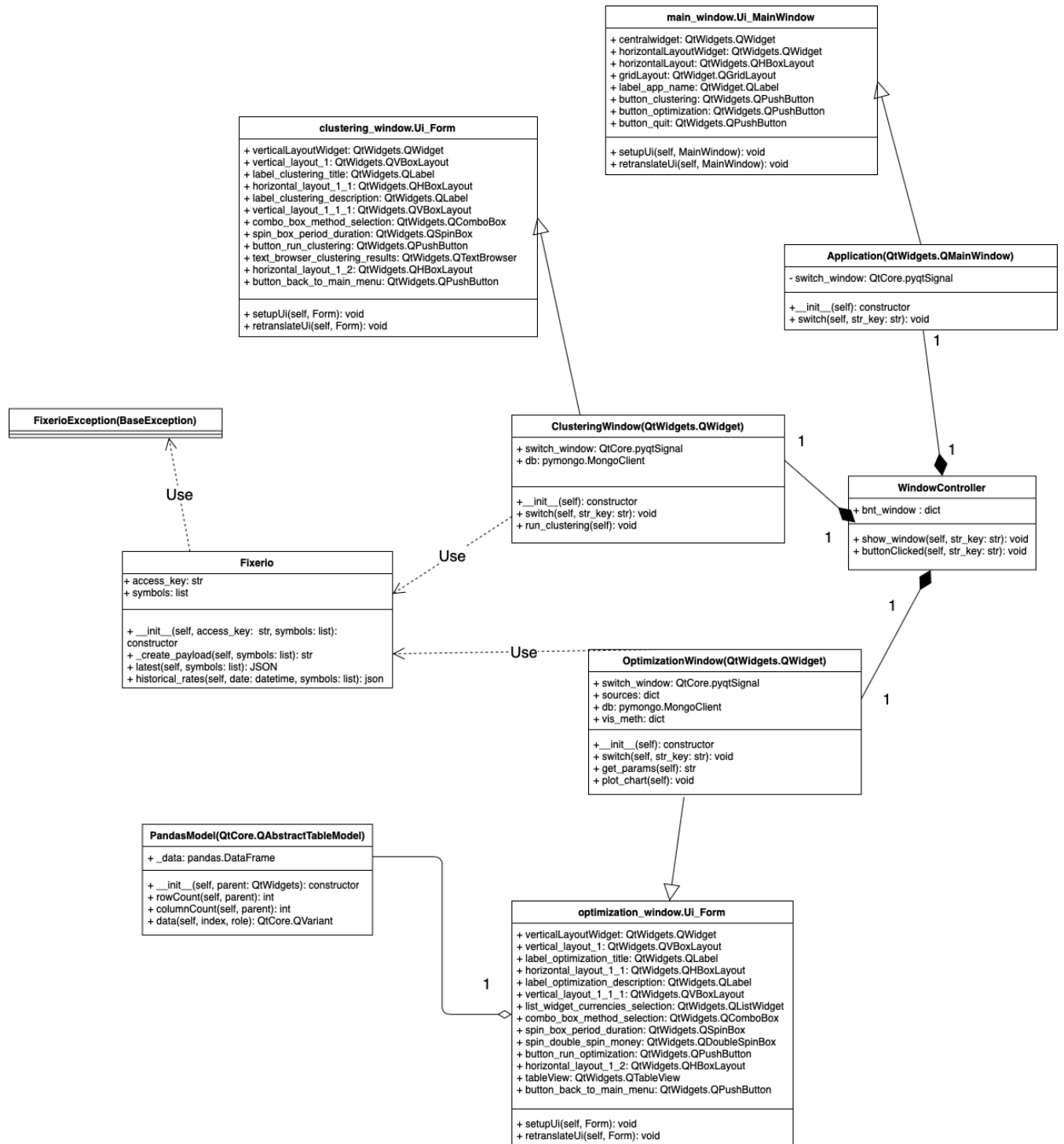


ДОДАТОК В ДІАГРАМА КЛАСІВ

Діаграма класів

*Програмне та математичне забезпечення оптимізації
портфелю активів на ринку іноземних валют*

(Найменування програми (документа))



Ім'я користувача:
Попенко Володимир Дмитрович

ID перевірки:
1005442012

Дата перевірки:
12.12.2020 23:26:58 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2020 23:34:28 EET

ID користувача:
77149

Назва документа: Jasenova_magistr_ip92mp

Кількість сторінок: 61 Кількість слів: 8967 Кількість символів: 68895 Розмір файлу: 2.02 MB ID файлу: 1005732774

12% Схожість

Найбільша схожість: 4.52% з Інтернет-джерелом (https://cad.kpi.ua/attachments/475_Petrishenko_magistr.pdf)

8.14% Джерела з Інтернету 164 Сторінка 63

11.9% Джерела з Бібліотеки 641 Сторінка 65

2.19% Цитат

Цитати 5 Сторінка 66

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 12